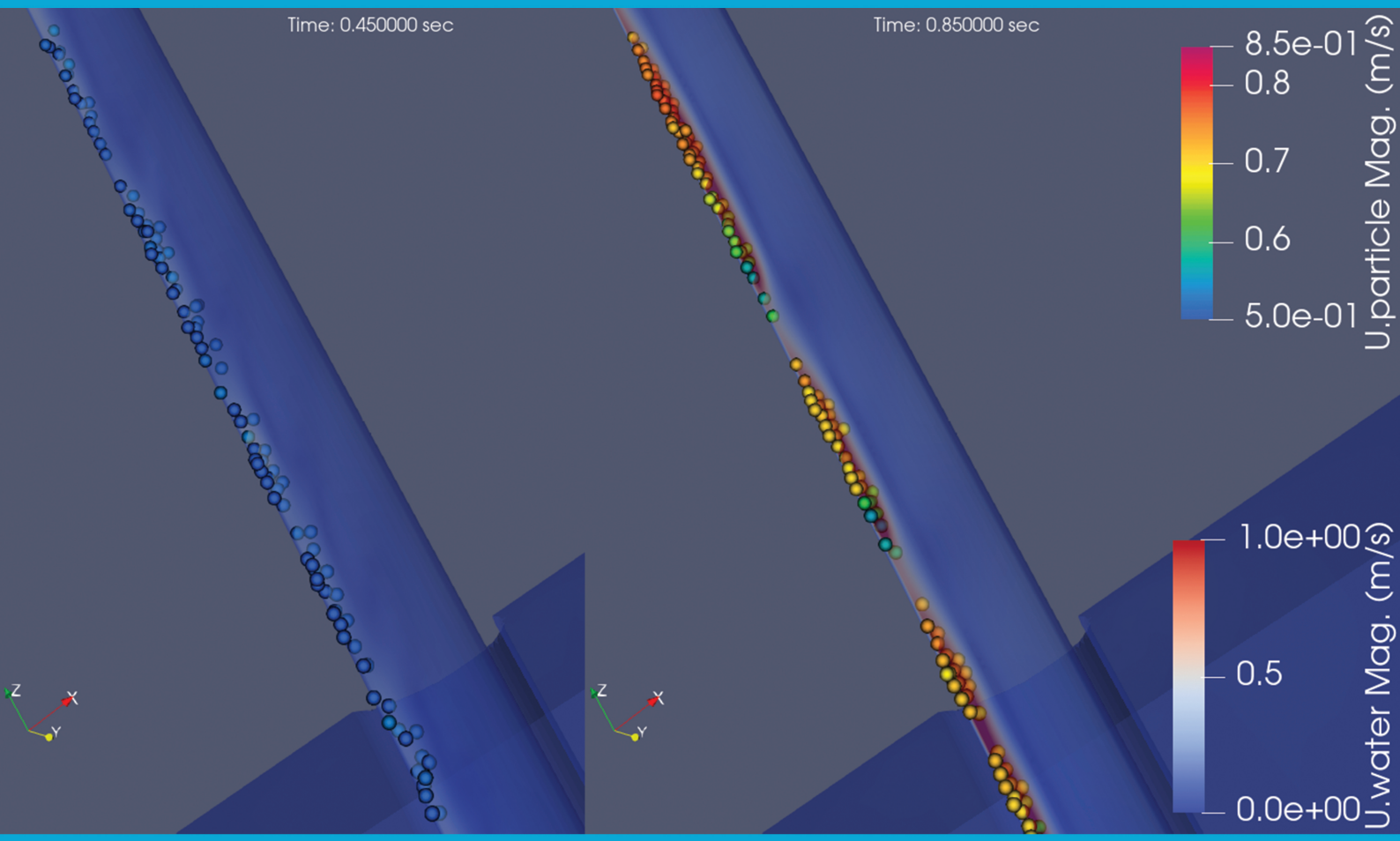
MSc thesis in Offshore & Dredging Engineering

Semi-resolved CFD-DEM simulations of the rock behavior inside an inclined fall pipe

M.H. Verhoog
2024



Boskalis

TU Delft

MSc Thesis in Offshore & Dredging Engineering

Semi-resolved CFD-DEM simulations of the rock behavior inside an inclined fall pipe

M.H. Verhoog

June 2024

A Thesis submitted to the Delft University of Technology in partial fulfillment of the requirements for the degree of Master of Science in Offshore & Dredging Engineering

M.H. Verhoog: *Semi-resolved CFD-DEM simulations of the rock behavior inside an inclined fall pipe* (2024)

Committee:	Dr.ir. Geert Keetels	TU Delft
	Dr.ir. Rudy Helmons	TU Delft
	ir. Remmelt van der Wal	Boskalis
	ir. Roeland Neelissen	Boskalis

Abstract

With the shift to renewable energy becoming more important than ever before and offshore wind farms being one of the sustainable energy resources available, many offshore wind farms are currently being built.
One of the difficulties of these wind farms built with monopiles is that the monopiles and electricity cables must be protected from scouring and falling objects. The protection is usually performed by placing a layer of rocks around and over these objects. For deep water up to 400 meters, vertical fall pipes are used to place the rocks directly under the rock placement vessel. This is impossible when placing rocks around monopiles, as the monopile stands in the way of where the vessel would have to lie during the rock placement procedure.
The inclined fall pipe (IFP) was introduced recently to combat this issue. The IFP, which hangs diagonally next to the vessel, can place rocks a couple of meters to the vessel's side, leaving a safe distance between the vessel and the monopile. Rocks tend to form clusters inside the pipe during placement with an IFP. This means the rocks exit the pipe in bursts of clusters, resulting in uneven rock layers if the rocks are placed during the movement of the vessel. Another result is less accurate placement of the rocks as the rocks at the outer layer of a cluster, during the free fall, are pushed further out, away from the intended placement location.

This thesis investigates the processes of rocks moving inside an inclined fall pipe and simulates this process numerically. This is achieved with a Finite-Volume solver for the fluid coupled with a Discrete Element model for the particles. The results are compared to existing experimental data.

The simulation results show the same rock behavior as in the lab tests for angles up to 60 degrees. The clusters start forming from the small volume concentration differences within the rock layer. The higher densities concentrate and become clusters, while the lower densities become gaps. In the simulations, this process is caused by the differences in the flow velocity of the fluid and the resulting drag differences of the particles. This knowledge is a first step towards finding solutions to lower the formation of clusters and thus, the more accurate placement of the rocks on the sea floor.

Furthermore, multiple points of improvement for the solver have been found. For angles of 75 degrees and steeper, the rock movements of the simulations miss a certain chaoticness, which may be attributed to missing turbulent drag forces in the solver. The velocities predicted by the solver are too high; multiple possible causes are speculated. The simulation results follow the same trends as the lab results, meaning they could be used for qualitative comparisons. Various additions to enhance the solver are proposed.
The influence of a system parameter, the distance factor was noticed to be stronger in shear flows than in homogeneous flows during a system parameter variation. This distance factor relates the fluid-particle interaction radius to the particle diameter. Finding limits for this parameter in shear flow could enhance the solver's prediction capability.

With the shift to renewable energy, more monopiles will be placed. The simulations show which forces cause the cluster formations and can be used to investigate ways of reducing them. This can reduce the spillage of rocks during operation, making the entire rock placement more efficient and will reduce costs.
The solver with the enhancements for better prediction capabilities could be used to further optimize the rock placement accuracy and efficiency. This would reduce wind farm costs and help the shift to renewable energy.

Acknowledgements

I would like to thank Geert Keetels, Remmelt van der Wal, Roeland Neelissen, and Rutger Voeten for their guidance while I was working on this thesis, for thinking with me about the issues I faced, and for showing me the right directions.
Many thanks go to Bas Nieuwboer for the multiple hours he spent helping me understand the solver, check the issues I faced with it, and finally, together with me, find out why it did not work as it was supposed to.
I am very grateful for Mohamed Elerian's support with all my OpenFOAM questions and the days I could spend at his university office troubleshooting my simulations.
Furthermore, I appreciate the kindness of all the colleagues of Boskalis for all the talks at the coffee machine. I especially appreciate Ebi for all the times we could have a talk, especially after work, and Saghi for all the kind words and lovely Persian sweets.

Then I want to thank my mother, father, grandma, Linda, and the rest of the family and my friends for always believing in me, even after so many years. Thank you, Castor and Marjolein, for always having your home available to study, Alma for wanting to get some fresh air with me at all times of the day, and Roebin for giving me the coffee machine and coffee talks I needed to finish this thesis.
Finally, I want to thank Meike for staying up with me when I was writing, checking my writing afterward to find every spelling mistake, and always being ready when needed. I could not have done it without you.

Contents

List of Figures

List of Tables

Nomenclature

α_c	Continuous phase volume fraction [-]
$\alpha_{p,j}$	Particle volume fraction [-]
$\bar{\bar{\tau}}$	Effective stress tensor, including both the viscous and turbulent shear stresses acting on the fluid $[N/m^2]$
β	Hindered settling exponent for the drag force [-]
β_L	Coefficient for the relation of Legendre to compare the wet and dry restitution coefficient, 35
Δt	Lenght of a timestep [s]
Δx	Size of the grid cells [m]
ϵ_{dry}	Restitution coefficient for dry particles [-]
ϵ_{wet}	Restitution coefficient for wet particles [-]
Γ	Dimensionless angular velocity [-]
γ	Distance factor, relating the standard deviation of the kernel function to the particle diameter [-]
μ_f	Sliding friction coefficient [-]
μ_r	Rolling friction coefficient [-]
ν	Laminar kinematic viscosity magnitude $[m^2/s]$
$\phi(r_{j,n})$	Kernel for mapping the particle information of particle n to cell n of the Eulerian mesh $[m^{-3}]$ (for a 3-dimensional simulation)
$\psi_{2D(rj,n)}$	Kernel function in a 2D case for mapping the Eulerian information of cell j to the particle n $[m^{-2}]$
$\psi_{3D(rj,n)}$	Kernel function in a 3D case for mapping the Eulerian information of cell j to the particle n $[m^{-2}]$
ρ_c	Density of the continuous fluid $[kg/m^3]$
ρ_p	Density of a particle $[kg/m^3]$
σ	Standard deviation of the Gaussian kernel [m]
$\vec{\omega}$	Angular velocity of a particle $[rad/s]$
$\vec{c}_j$	Location of the cell center of cell j [m]
$\vec{c}_{p,n}$	Center of particle n [m]
$\vec{F}_b$	Buoyancy force [N]
$\vec{F}_g$	Gravity force [N]
$\vec{F}_L$	Lift force [N]
$\vec{F_M}$	Magnus lift force [N]

$\vec{F_{AM}}$ Added mass force [N]

$\vec{F_{d,0}}$ Drag force on a single particle in the absence of surrounding particles [N]

$\vec{F}_d$ Drag force [N]

$\vec{F}_f$ Sliding friction force [N]

$\vec{F}_i$ Interaction force between the fluid and the discrete particles [N]

$\vec{f}_i$ Interaction force of the discrete particles acting on the fluid per unit volume [N/m^3]

$\vec{F}_N$ Normal force [N]

$\vec{F}_r$ Rolling friction force [N]

$\vec{F}_{pg}$ Pressure gradient force [N]

$\vec{F}_{visc}$ Viscous force [N]

$\vec{g}$ Gravitational constant, 9.81 m/s^2

$\vec{u}_p$ Velocity of the particle [m/s]

$\vec{u}_{clp}$ Undisturbed fluid velocity at the particle location [m/s]

$\vec{u}_{n,in}$ Velocity of a particle perpendicular to a wall before collision [m/s]

$\vec{u}_{n,out}$ Velocity of a particle perpendicular to a wall after collision [m/s]

A_p Projected area of the particle facing the flow direction [m^2]

C_d Drag coefficient [-]

C_L Lift coefficient [-]

C_M Magnus lift coefficient [-]

C_{AM} Added mass constant, 0.5

D_n Nominal diameter [m]

d_p Diameter of a particle [m]

D_{50} Median sieve diameter [m]

D_{s50} Median diameter of spheres with a equal volume to the rocks [m]

E Young's modulus [N/m^2]

f Vortex frequency [Hz]

L Length of the pipe [m]

m_p Mass of a particle [kg]

N Amount of particles [-]

n_d Number of spatial dimensions of the simulation [-]

$N_{timesteps}$ A user defined number of timesteps in which a collision is calculated [-]

p Pressure acting on the continuous phase [N/m^2]

R Limit radius of a domain [m]

r_p	Radius of a particle [m]
$r_{j,n}$	Distance from the cell j to the center of particle n [m]
Re_p	Particle Reynolds number depending on just the particle velocity [-]
Re_s	Particle Reynolds number based on the slip velocity [-]
S_f	Corey shape factor [-]
St	Strouhal number [-]
St_{AM}	Stokes number based on the inertia of the particle including the added mass of the water [-]
t	Time [s]
V	Reference volume, typically the volume of a mesh cell [m^3]
V_p	Volume of a particle [m^3]

1 Introduction

This chapter discusses the practical uses of the inclined fall pipe and the challenges of the traditional vertical fall pipe it overcomes. It then details the disadvantages of the stone behavior in the inclined fall pipe and the complexities of the numerical modeling of the rock placement process. After which, the objectives of the thesis will be represented, including the main and sub-questions answered in this thesis.

1.1 The inclined fall pipe (IFP)

Placing rocks on the seafloor is essential to protect offshore equipment and materials. The rocks protect vital sub-sea components from falling or dragging objects and protect the sand around them from the scouring effect. Traditionally, vertical fall pipes have been used for this purpose. However, in the relatively shallow waters where monopiles are placed, the use of vertical fall pipes becomes impractical. The vertical fall pipes are designed to deposit material directly underneath the vessel and can only move a couple of meters to the sides per hundreds of meters of depth.

A recent development is the inclined fall pipe (IFP). This pipe is mounted on the side of a rock placement vessel, allowing it to hang under an angle of 30 to 75 degrees compared to the flat water surface. In figure 1.1, the IFP of the Rockpiper can be seen.

Figure 1.1: IFP of the Rockpiper, depositing rock next to a monopile. [29]

The angle of the pipe and the mounting location on the side allow the IFP to deposit the rocks at a greater horizontal distance from the vessel than possible with a vertical fall pipe. Using the IFP, the vessel can keep a safe working distance from the upper section of the monopile and place the rocks on the seafloor next to or against the monopile.

1.2 Problem description

The rock behavior in the IFP differs greatly from that of that in the vertical fall pipe despite involving the same water and rocks.

In the vertical fall pipe, the rocks descend with some chaotic movement, spreading randomly over the pipe's cross-section. Depending on whether water entries are present in the side of the pipe, the water either flows downward along with the rocks or has minimal vertical movement.

The behavior of the rocks in the IFP is quite different. Due to the angle of the IFP, the rocks gather along the lower side of the pipe, sliding down the pipe wall toward the pipe's exit. The water flows alongside the sliding rocks, but only where rocks are present. In the parts of the pipe with few or no rocks, a counterflow is present. This counterflow supports the clustering process of the rocks, which, in turn, negatively influences the precision of the placements of the rocks on the seafloor. Although some studies have explored rock behavior in and below the IFP through laboratory model tests (e.g. Fontijn [30]) and numerical simulations of rock dispersion after the IFP exit (Hogenhuis [31]), there is a lack of numerical simulations specifically addressing the behavior of the rocks inside the IFP. One of the challenges in modeling this process is the importance of both rock-rock and rock-water interactions, which cannot be handled by current numerical methods without simplifications at this scale and in a reasonable time.

For this thesis, it has been chosen to work with the CFD-DEM solver created by Bas Nieuwboer, called bigGaussDPMFoam [27]. This solver uses a semi-resolved CFD-DEM approach, an approach that falls between the resolved approaches, where forces on the particles are derived by solving the fluid-particle interaction in detail, and the unresolved approaches, where the interaction forces are calculated using the drag formulation and the fluid velocity of the grid cell the particle is residing. In the unresolved approach, the fluid movement is slowed down by the interaction force, but local velocity differences in the size of the particle are not possible. The semi-resolved solver has the best of both the resolved and unresolved approaches. The solver is fast due to the use of the drag formulation for the fluid interaction and the use of a kernel function (more about this in chapter 4), which takes an average of the fluid velocity from the cells around it, resulting in higher accuracy for the fluid-solid interaction forces and the possibility for local flow differences. Due to the scale of the simulations, with a couple -of hundreds of particles, and the importance of local flow differences for the individual particle movements, a semi-resolved solver is preferred. The bigGaussDPMFoam solver was chosen since semi-resolved solvers are not widely available, and this one is available to Boskalis and the TU Delft. Due to restrictions on the solver, it simplifies the particles as spheres, and not all the particle physics is included. One of the challenges is to find out to which extent the solver can simulate the processes inside the inclined fall pipe.

1.3 Research objective

As the problem description states, the solver used for simulating the inclined fall pipe process is not perfect. Some of the simplifications and missing physics can influence the simulation quality. Hence, the main research question is as follows:

To what extent can the rock behavior inside the IFP be predicted using a semi-resolved CFD-DEM method?

With the sub-questions:

1. Do the simulations predict the same cluster formation and flow effects as in model lab tests?
2. To what extent can the solver predict the rock velocity and production for different angles of the inclined fall pipe and different mixture concentrations in the IFP?
3. What is the influence of various system parameters?
4. How can the prediction capability of the solver be improved?

2 Inclined Fall Pipe

2.1 Overview

A schematic drawing of the rock dumping process with an inclined fall pipe can be seen in figure 2.1. In the figure, the entire rock placement process is divided into the following six sub-processes.

1. Rock inlet
2. Rock immersing in the water
3. Rock sliding/falling inside the pipe
4. Rock exiting the pipe
5. Rock in the free water column
6. Rock colliding with the seabed

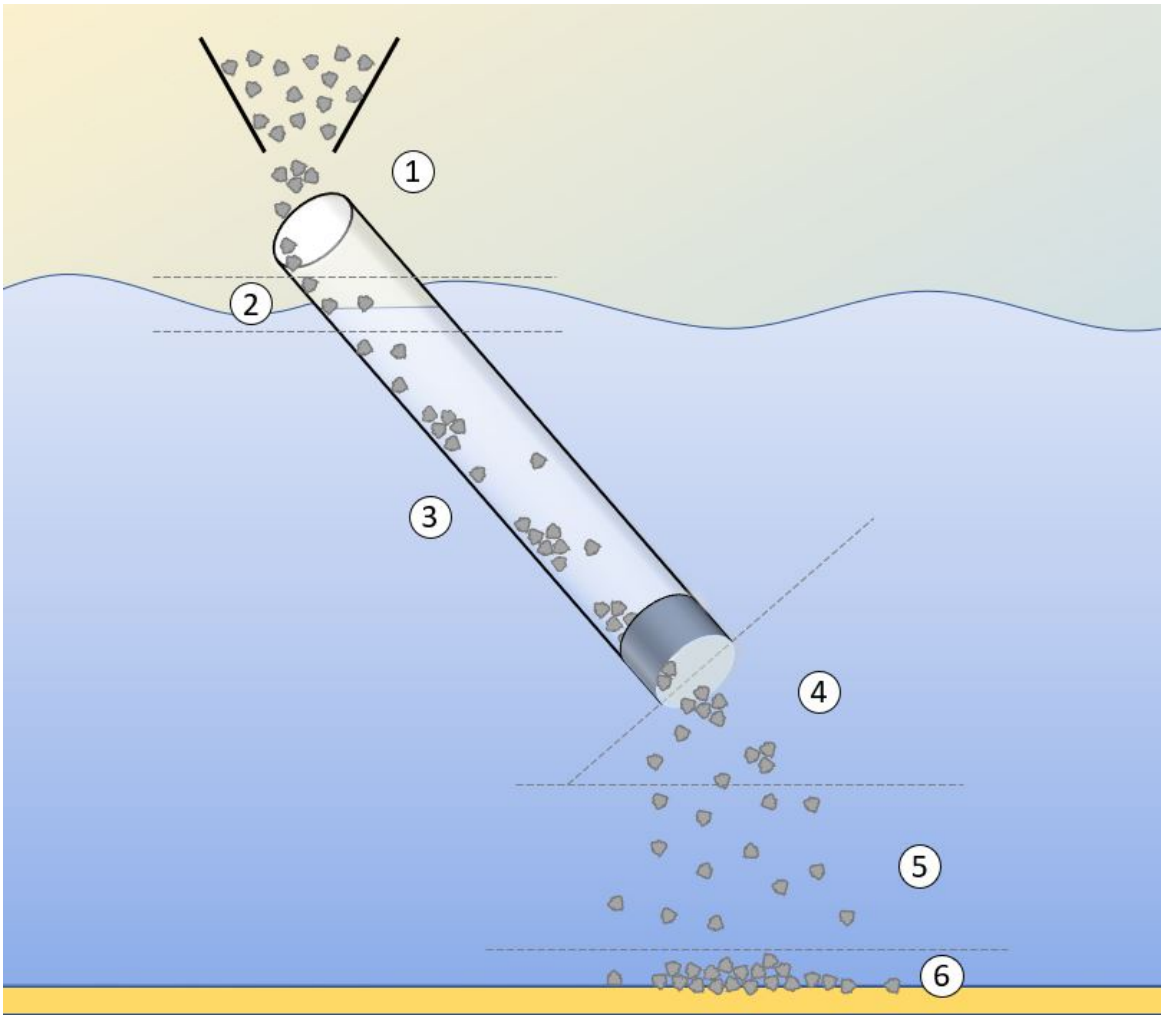


Figure 2.1: Sketch of an inclined fall pipe with its different sections and sub-processes.

Each sub-process will be discussed briefly in its own section, including which research has been performed into these processes.

1. Rock inlet

The rock inlet consists of a vibratory feeder or conveyor belt and a funnel directing the rocks into the pipe. This part of the process is above the water level, which limits the contributing forces to just interaction forces between the rocks, the rocks with the funnel and some air resistance. This part of the process does not differ from other fall pipe systems or transport systems on land. Research into this process by Herrman et al. [8] and Raafat et al. [10] shows that after leaving a hopper, particles will start to form density waves. The time and distance in the IFP before reaching the water surface is, however, too little for these waves to form.

2. Rock immersing in the water

The sub-process of rocks immersing in water starts at the water's surface and ends just below it. The rocks move through the waterline, causing some water to be entrained in their movement, speeding up the water and slowing down the rocks. Air will be pulled along in the water, form bubbles, and flow back to the water surface. This sub-process does not differ from vertical fall pipes or other processes in which rocks are released above the water level.

3. Rock sliding/falling inside the pipe

The area for the sub-process of rock sliding and/or falling in the pipe begins just below the water surface and goes up to the pipe end. This sub-process is researched in the form of model lab tests by Fontijn [30].

According to Fontijn's research, the rock behaves differently depending on the angle of the pipe, the production rate of the rocks, and the rock sizes. In all cases for inclined pipes, the rocks will move near or along the lower side of the pipe, forming a rock layer moving down and a water layer above it flowing up. This layer will be more continuous for larger productions with smaller rock sizes, while the rocks in the layer will form clusters with larger rock sizes or smaller productions.

4. Rock exiting the pipe

Just outside of the pipe, near its exit, the rocks are not supported anymore by the pipe wall but are still influenced by the flow from the pipe. This sub-process is researched with model lab tests by Fontijn [30] and numerical modeling by Hogenhuis [31]. Fontijn researched this process together with the next process by researching the horizontal spread and offset of the stones for different angles, production rates, and rock sizes. Hogenhuis [31] used DriftFlux (CFD) to model the flow and turbulence field below the pipe and combined this with a numerical model for the induced lift force by vortex shedding and the influence of turbulence. Each rock's movement was calculated individually, with starting positions on the end of the pipe dependent on the DriftFlux concentration field. The influence of turbulence on the rocks for vertical fall pipes was researched by Kevelam [25] for vertical fall pipes.

5. Rock in the free water column

The sub-process of rock falling through the free water column starts when the pipe's flow field no longer influences the rock movement. The lift force due to vortex shedding influences the movement of the rocks in this part of the process. Hogenhuis's research also included this sub-process. Earlier research about the movement of rocks in water has been done by Cregten [9], who described the Single Stone model for the movement of rocks in stagnant water, and Van der Wall [15] described the behavior of falling groups of stones.

6. Rock colliding with the seabed

After falling through the water column, the rocks reach the seabed. The rocks can roll, slide, bounce, or sink into the sea bed depending on the velocity and soil type. Beemsterboer [24] described the effect the interaction with the seabed has on volume loss of the placed rock layer.

2.2 Defining the scope

The sub-processes of the inclined fall pipe rock placement process and the research into these objects are discussed in the section above. This research will focus on numerically modeling sub-process 3, the process of rocks moving through the pipe, as highlighted in figure 2.2. The sub-processes 1 and 2 are the same for IFP's as for vertical fall pipes, while the interest of this thesis lies in finding the cause of differences with the vertical fall pipe process. Sub-process 6 is highly dependent on the seabed properties. Sub-processes 4 and 5 are modeled in Hogenhuis' research [31]. For this research, the process in the pipe was simplified with a constant density flow model of DriftFlux, while the lab tests of Fontijn [30] showed that the rocks tend to form clusters inside the pipe. The main goal of this research is to model this clustering process inside the IFP correctly.

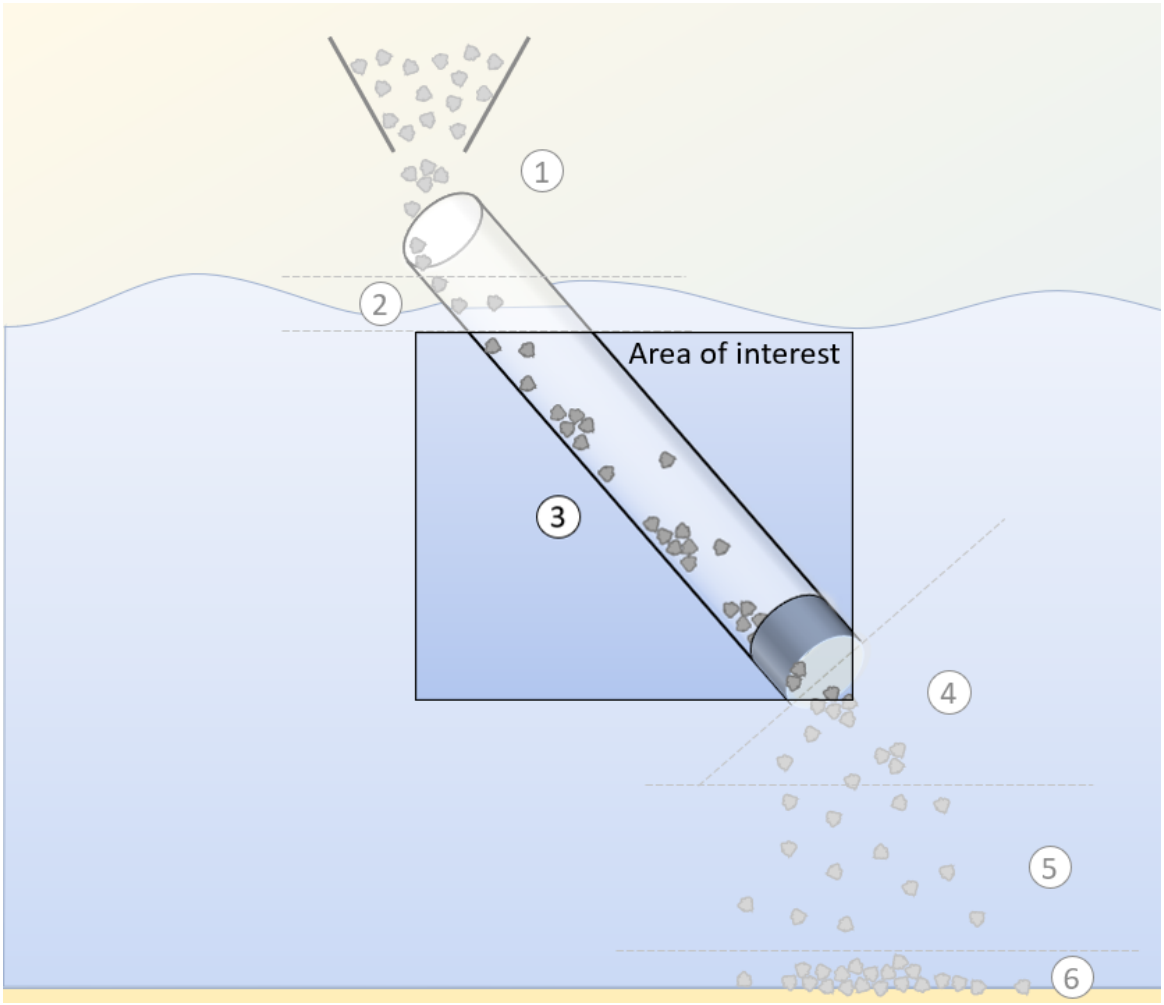


Figure 2.2: The area of interest for the numerical research.

2.3 Dimensions of the IFP

The dimensions of the domain for the numerical simulations will be based on those of Fontijn's model lab results [30], so the simulation results can be compared to those from the lab. The model's dimensions are scaled 15:1 compared to the IFP used on Boskalis' Rockpiper. During the scaling, the Froude number was taken as a constant, which resulted in a scale factor of $15^{2.5}$ for the production. The prototype and scale-test dimensions overview are in table 2.1.

	IFP	*Scale model*	*Scale factor*
Pipe length	21.15 m	1.41 m	15
Pipe diameter	1.8 m	0.12 m	15
Production	2000 t/h	0.64 kg/s	$15^{2.5}$
	1500 t/h	0.48 kg/s	$15^{2.5}$
	500 t/h	0.16 kg/s	$15^{2.5}$
Rock diameter	0.12 - 0.25 m	8 - 16 mm	15
	0.28 - 0.48 m	19 - 32 mm	15

Table 2.1: Table of dimensions for the prototype on the Rockpiper and the scale tests of Fontijn. [30]

3 Particle Physics

This chapter provides a step-by-step explanation of the physics of particles in water, beginning with the simplest scenario, a round particle surrounded entirely by water. After this, the interaction with a (pipe) wall is introduced. Lastly, the differences in these forces with non-round particles (rocks) will be identified.

3.1 Physics of a single particle in fluid

3.1.1 Gravity and buoyancy

A single immobile spherical particle in still-standing water with a uniform pressure field experiences just 2 forces. One is the gravity force ($\vec{F}_g$) pulling the particle down, and the other is the buoyancy force ($\vec{F}_b$) pushing the particle up. The gravity force is the mass of the earth pulling on the particle by the gravitational constant ($\vec{g}$, a vector with a magnitude of 9.81 m/s^2 directed towards the earth) times the particle mass (m_p) (or its volume (V_p) times its density (ρ_p)). The buoyancy force results from gravity pulling down the water the particle is immersed in. This force equals $\vec{g}$ times the volume it inhabits in the water times the density of the water (ρ_c). The equations for both forces are given in equation 3.1 and 3.2.

$$\vec{F}_g = \rho_p \, \vec{g} \, V_p \tag{3.1}$$

$$\vec{F}_b = -\rho_c \, \vec{g} \, V_p \tag{3.2}$$

Due to the similarity of the equations, it can easily be concluded that $\vec{F}_g$ and $\vec{F}_b$ result in a net upward force when the density of the particle (ρ_p) is smaller than the density of the water (ρ_c), a net downward force when ρ_p is larger than ρ_c and a net zero force when the densities are equal to each other. In the case of stone particles, the density of the particle is higher than the density of the water, so a particle at rest will experience a force down and start accelerating downwards.

3.1.2 Viscous drag

The same spherical particle, but now moving with a constant velocity, experiences a drag force ($\vec{F}_d$) besides the $\vec{F}_g$ and $\vec{F}_b$. In short, this force is exerted on the particle by the viscous fluid moving around the particle, slowing the particle down. The direction of the drag force is always in the opposite direction of the particle velocity. In the case the water also moves, the drag force is in the direction of the slip velocity ($\vec{u}_{c|p} - \vec{u}_p$). The drag force relation is given for round particles in equation 3.3.

$$\vec{F}_d = \frac{1}{2} \, C_d \, A_p \, \rho_c \, (\vec{u}_{c|p} - \vec{u}_p) \, |\vec{u}_{c|p} - \vec{u}_p| \tag{3.3}$$

With:
C_d as the drag coefficient [-]
A_p as the projected area of the particle facing the flow direction [m^2]
$\vec{u}_p$ as the velocity of the particle [m/s]
$\vec{u}_{c|p}$ as the undisturbed fluid velocity at the particle location [m/s]

A free-falling particle in water will keep accelerating until the drag force is as large as the difference between the gravity and buoyancy force, $|\vec{F}_d| = |\vec{F}_g| - |\vec{F}_b|$. This is called the terminal velocity of the particle. The terminal velocity will be used for validation steps in section 5.2.1.

The drag coefficient (C_d) in equation 3.3 is related to the particle Reynolds number based on the slip velocity, which is the difference between the fluid velocity at the location of the particle and the particle velocity. The Reynolds number is defined as in equation 3.4.

$$Re_s = \frac{|\vec{u}_{c|p} - \vec{u}_p|\ d_p}{\nu} \tag{3.4}$$

With:
Re_s as the Reynolds particle number based on the slip velocity [-]
d_p as the particle diameter [m]
ν as the laminar kinematic viscosity magnitude [m^2/s]

Many formulations for the drag coefficient exist, all based on empirical relations. The drag coefficient is divided into three regions: a laminar region or Stokes regime ($Re_s < 1$), a transition region or intermediate regime ($1 < Re_s < 2000$), and a turbulent region or Newton's regime ($Re_s > 2000$). In the laminar region, the drag coefficient depends on the particle Reynolds number by Stokes law $C_d = 24/Re_s$. In the turbulent regime, up to the particle Reynolds number of $2 \cdot 10^5$, the drag coefficient is about constant near a value of 0.4 to 0.46. In between is the transition region. The value drops suddenly for Reynolds numbers above $2 \cdot 10^5$. This is called the drag crisis. The Re_s of the simulated particles will stay below the $2 \cdot 10^5$

The solver created by Nieuwboer [27] uses the drag relation from Brown and Lawler [16] (equation 3.5), which is valid for particle Reynolds Numbers below $2 \cdot 10^5$.

$$C_d = \frac{24}{Re_s}\ (1 + 0.150\ Re_s^{0.681}) + \frac{0.407}{1 + 8710\ Re_s^{-1}} \qquad Re_s \leq 2 \cdot 10^5 \tag{3.5}$$

The above relation is valid for spherical particles fully surrounded by fluid. Section 3.2.5 gives the differences in the C_d values found for particles moving along pipe walls. Section 3.4.1 gives C_d values for non-spherical particles.

3.1.3 Added mass

During acceleration or deceleration, a particle will experience an added mass force $\vec{F}_{AM}$ in addition to the previously mentioned forces. This force works against the direction of the acceleration or deceleration. It can be attributed to the volume of water surrounding the particle that needs to be accelerated or decelerated along with the particle. The relation of the added mass force as used in the solver of Nieuwboer [27] can be found in equation 3.6. The constant C_{AM} is usually taken as 0.5.

$$\begin{aligned} \vec{F}_{AM} &= C_{AM}\ V_p\ \rho_c\ (\frac{D\vec{u}_{c|p}}{Dt} - \frac{d\vec{u}_p}{dt}) \\ \frac{D\vec{u}_{c|p}}{Dt} &= \frac{d\vec{u}_{c|p}}{dt} + \vec{u}_{c|p} \cdot \nabla \vec{u}_{c|p} \end{aligned} \tag{3.6}$$

3.1.4 Pressure gradient and viscous forces

Considering a particle standing still in space but with the fluid around it moving. When the undisturbed fluid flow is uniform on all sides of the particle, it is as if the particle is moving and the water is standing still. The drag force will suffice in that case. However, when the flow is non-uniform, like in a rotating flow, more forces will come into play. These are the pressure gradient force and viscous force.

According to Maxwell and Riley [4], the pressure gradient force, together with the viscous force, is equal to the total derivative of the fluid velocity. This means the forces can be calculated using equation 3.7.

$$\vec{F}_{pg} + \vec{F}_{visc} = V_p\ \rho_c\ \frac{D\vec{u}_{c|p}}{Dt} \tag{3.7}$$

3.1.5 Saffman lift force

Small particles in a shear flow experience a lift force perpendicular to the direction of this shear flow. This force is called the Saffman lift force, and according to the research of Saffman [3], is for spherical particles with small particle Reynolds numbers ($Re_s \ll 1$). This force is not included in the solver. It can be neglected since the particles will have Reynold numbers above 1.

3.1.6 Magnus lift force

Rotating particles in a uniform flow experience a force perpendicular to the flow called the Magnus lift force. This is due to the particle slowing the flow down on one side and speeding the flow up on the other by turning. This results in a lower pressure on the faster side and a higher pressure on the slower side. In figure 3.1, the combination of rotating, flow, and force directions are shown. The force of the Magnus effect per unit of volume for a sphere is defined by Rubinov and Keller [2] and Nigmatulin [5] as shown in equation 3.8.

$$\vec{F}_M = C_M \, \rho_c \, [\vec{\omega} \times (\vec{u}_{c|p} - \vec{u}_p)] \tag{3.8}$$

With:
$\vec{\omega}$ as the particle's angular velocity $[rad/s]$
C_M as the Magnus force coefficient.

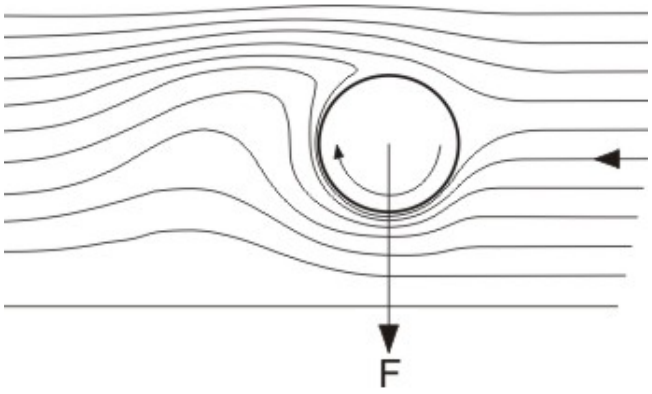


Figure 3.1: The resulting Magnus force direction from a rolling particle in a uniform flow field. [32]

The Magnus force coefficient, C_M, depends on the dimensionless angular velocity, Γ, as given in equation 3.9. According to Lukerchenko et al. [23], for values of Γ below 3.5, the C_M varies from 0.023 to 0.048 and $C_M = 0.07/\Gamma$ for larger values of Γ, $5 < \Gamma < 12$.

$$\Gamma = r_p \, \frac{|\vec{\omega}|}{|\vec{u}_{c|p} - \vec{u}_p|} \tag{3.9}$$

With:
r_p as the particle radius [m]

The used solver does not include the Magnus lift force, as in Nieuwboer's research [27], the Magnus lift force was negligible compared to the rock cutter's rotary and pressure difference forces. It is expected that the Magnus lift force does have a significant influence on the particles since no rotational or pressure difference driving forces are acting on the IFP system, leaving the weaker gravity as main driving force.

The magnitude of this force can be estimated by taking the angular velocity of the particle while rolling along the tube wall. Since the friction with the wall is the main driving force for the rolling motion, this will give a maximum for spherical particles. Rocks will have lower angular velocities as they will slide more due to their shape. The particles in the lab tests at 75 degrees move with about 0.8 m/s. When the particle is rolling and not slipping along the wall, the ratio between the particle velocity and the angular velocity will be $1/r_p$. This means the angular velocity will be around $0.8/r_p$ rad/s. Assuming the fluid does not move gives $\Gamma = r_p \, \frac{0.8/r_p}{0.8-0} = 1$.
When taking equation 3.8 and using 0.3 for C_M, 1000 for ρ_c in, the above-mentioned velocities, and a particle

with a radius of 7 mm, this gives $|\vec{F}_M| = 0.03 \cdot 1000 \cdot 0.8/0.007 \cdot 0.8 = 2800\ N$. The 7mm particle, with a density of 2650 kg/m^3, has a mass of $2650 \cdot 4/3 \cdot \pi \cdot 0.007^3 = 0.0038$ kg. The acceleration of the particle due to the Magnus force will be $2800/0.0038 = 7 \cdot 10^5\ m/s^2$. This example is the upper limit, calculated for spheres instead of rocks, where rocks are expected to have significantly lower angular velocities. In this example in the IFP, the force is directed toward the wall, so it will be compensated with normal force from the wall, but it shows that the Magnus effect force can be very large compared to the gravitational acceleration of 9.81 m/s^2.

3.1.7 Basset history force

Like the added mass force (section 3.1.3), the Basset history force occurs in unsteady conditions. The Basset history force occurs when the particle accelerates fast compared to the fluid around it and is attributed to the lagging boundary layer development around the particle. This force can become relatively high for small particles with large accelerations. However, according to Rostami [18], the Basset history force is less significant for large particle Reynolds numbers. For values between 1000 and 5000, the error for neglecting the Basset history force is only about 1% to 4.3%. The force is not included in the used solver. The error will be neglected as the terminal velocity of the particles has particle Reynolds numbers above 5000.

3.1.8 Turbulent dispersion force

A particle inside a turbulent flowing fluid will be moved by the turbulent eddies of the fluid. Since the eddies will change direction over time, the force direction will also change, which can appear to be seemingly random. Some CFD solvers include models to simulate these movements, like the StochasticDispersionRAS model of OpenFOAM [33] and the Random Walk model of Ansys [28]. These models use a random value to set a direction in which the eddy will move the particle. The turbulent eddy affects the particle for the time the particle moves through the eddy, and the eddy has not dissipated. In the models, the pass-through time is calculated, and the lifetime of the eddy will be modeled with a random number. A new random direction will be chosen when either the eddy lifetime or pass-through time is over. This force and the model are not included in the used solver as this model is not directly compatible with the way the particle-fluid coupling works in this solver. The result is a lack of seemingly random movements of the rocks in the direction tangential to the slip velocity. To estimate the influence of these eddies, a calculation is performed with an eddy with the diameter of the pipe, which is the largest size possible in this scenario. The eddy lifetime will be assumed to be longer than the pass-through time. The turbulent dispersion forces for the cases simulated in the IFP can reach, in extreme cases, a maximum value of about 0.02N, assuming that the sideways velocity inside these eddies is not larger than 0.5 m/s. This equals an initial acceleration of almost 6 m/s^2. This acceleration rapidly decreases as the velocity difference declines. The maximum time the particles are influenced by the same eddy equals the pass-through time of 0.25 seconds.

3.1.9 Vortex shedding

When placing a sphere in a strong enough flow ($Re_s > 300$), the flow behind the sphere will form vortices that shed alternately, as visualized in figure 3.2.

The shedding of the vortices will introduce an alternating force on the sphere perpendicular to the flow. The vortex frequency (f) depends on the size of the particle, the velocity of the particle to the flow, and the dimensionless Strouhal number (St) (equation 3.10).

$$St = \frac{f\ |\vec{u}_{c|p} - \vec{u}_p|}{d_p} \tag{3.10}$$

The Strouhal number is related to the particle Reynolds number as seen in figure 3.3a from Sakamoto and Haniu [6]. The periodic vortex shedding occurs for Re_s between 300 and $3.7 \cdot 10^5$. A higher and a lower mode of Strouhal numbers exist for particle Reynolds numbers between 800 and $6 \cdot 10^4$. The lower mode is visible in detail in figure 3.3b.

To grasp how often these vortex sheds occur over the length of the lab model test pipe of 1.44 meters, the amount of shedding vortices is calculated for the velocities: 0.022, 0.2, 0.5, 0.7, 1.0, and 1.8 m/s. The particles

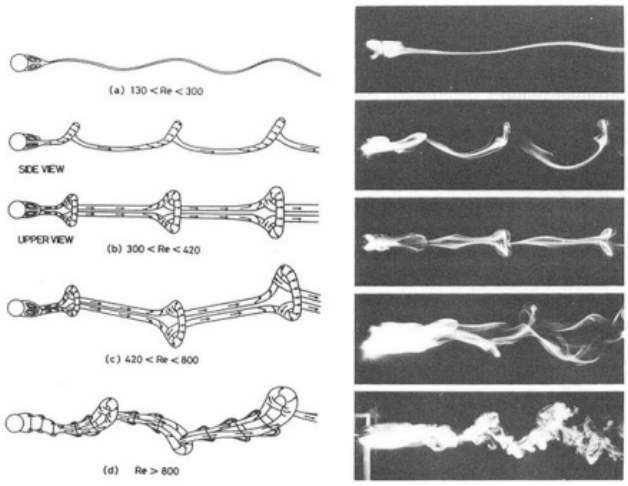
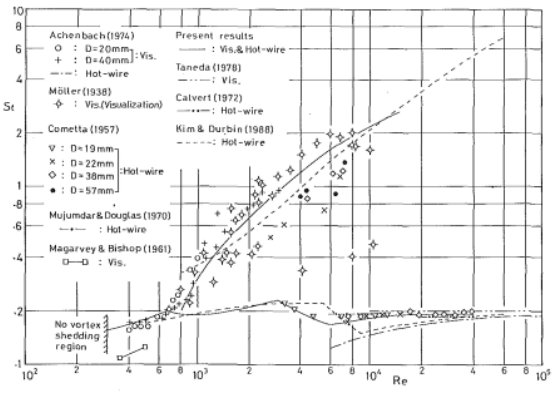
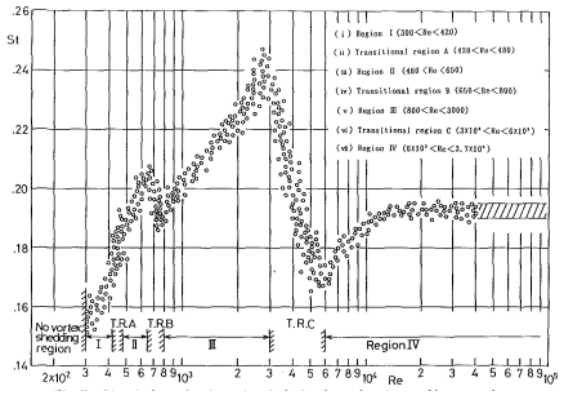


Figure 3.2: The vortices shedding from a sphere. [6]



(a) The Strouhal-Reynolds relation. Both the higher and lower modes. [6]



(b) The Strouhal-Reynolds relation only the lower mode. [6]

Figure 3.3: The Strouhal-Reynolds relation. Both the higher and lower mode (a) and only the lower mode (b).

are 14 mm in diameter, and the kinematic water viscosity (ν) is taken as $1.0034 \cdot 10^{-6} m^2/s$. The velocities are chosen as follows:

- 0.022 m/s is the velocity with a particle Reynolds number just above the minimum limit for vortex shedding.
- 0.2 m/s is the velocity with a particle Reynolds number for which the Strouhal-number peaks in the low mode.
- 0.5 m/s is the maximum settling velocity of a non-spherical particle sliding along the wall of a 75-degree pipe.
- 0.7 m/s is about the average settling velocity of multiple rocks in a pipe
- 1.0 m/s is estimated for one of the faster particles moving toward a part of the flow that does not move along with the particle
- 1.8 m/s is estimated for one of the fastest particles moving toward a part of the flow, which flows with maximum velocity against the particle direction.

The Strouhal numbers are estimated from both the figures 3.3a and 3.3b, the number of vortices shed is estimated as the shedding frequency times the time the particle will take to traverse the length of the tube. The results for both the higher and lower Strouhal-number modes can be seen in table 3.1.

Table 3.1: The amount of vortices shed over the pipe length for different possible velocities of the particles.

Slip Velocities (m/s)		0.022	0.2	0.5	0.7	1	1.8
Re_s		308	2800	7000	9800	14000	25200
Lower Strouhal mode	Strouhal number	0.16	0.24	0.18	0.18	0.19	0.19
	Number of vortices shed	16.5	24.7	18.5	18.5	19.5	19.5
Higher Strouhal mode	Strouhal number	0.16	0.3	1.6	2	2.5	4.2
	Number of vortices shed	16.5	30.9	164.6	205.7	257.1	432.0

The number of vortices for the lower mode stays relatively constant, around 15 to 25 over the pipe length. Vortices shed much more often for the upper mode at velocities of 0.5 m/s and above. About 200 to 450 times over the length of the pipe, which is about 10 to 20 times more often than for the lower mode. In figure 3.2, it can be seen that the vortices for the higher mode are smaller and thus result in smaller forces than those of the lower mode. The forces of the shedding vortices on the particle are known to be small compared to the slip velocity and often are only considered in situations where the vortices cause vibrations. The vortex-shedding forces are not included in the used solver. As a result, the particles experience less force in the direction tangential to the slip velocity. The forces would not move the particles significantly itself, but could start a motion that is enhanced by other factors. The vortices created can influence other particles as an eddy from the turbulent dispersion force (section 3.1.8).

3.2 Physics of a particle interacting with a wall

3.2.1 Normal force

At the moment of contact between a particle and a wall, the normal force ($\vec{F}_N$) is the force exerted by the wall on the particle perpendicular to the wall surface. This force equals the force from the particle on the wall perpendicular to the wall. The normal force is the force that prevents the particle from moving through the wall. When the particle rests on a horizontal wall, the force equals the difference in gravity and buoyancy force $\vec{F}_N = \vec{F}_g - \vec{F}_b$. If the particle is in contact with an inclined wall, a resultant force parallel to the wall will remain as the $\vec{F}_N$ only works perpendicular to the wall.

3.2.2 Sliding friction force

The sliding friction $\vec{F}_f$ is the force between the wall and the particle parallel to the wall. This force is non-zero when there is relative motion between the contact area of the particle and the wall. The sliding friction magnitude can be calculated by the product of a sliding friction coefficient (μ_f) and the normal force magnitude acting on the particle $|\vec{F}_f| = \mu_f \, |\vec{F}_N|$.

3.2.3 Rolling friction force

Rolling friction ($\vec{F}_r$) is like sliding friction, a force between the particle and the wall parallel to the wall. This force is non-zero when the contact areas of the wall and the particle have a relative motion of zero and the particle moves along the wall in a rolling motion. The rolling friction magnitude can be calculated by the product of a rolling friction coefficient (μ_r) and the normal force magnitude acting on the particle $|\vec{F}_r| = \mu_r \, |\vec{F}_N|$. The rolling friction force is usually much smaller than the sliding friction force.

3.2.4 Viscous collisions

Particles in a collision, either with a wall or another particle, have a different velocity before the collision ($\vec{u}_{n,in}$) than after the collision ($\vec{u}_{n,out}$). The ratio ($\vec{u}_{n,out}/\vec{u}_{n,in}$) between these velocities is called the restitution coefficient. The restitution coefficient for dry particles (ϵ_{dry}) is constant for the same materials. Particles in fluid colliding with a wall or with other particles experience a restitution coefficient (ϵ_{wet}) that is different than the one for dry particles. The restitution coefficient for particles in fluid is not constant for the same materials; it depends on the Stokes number.

The Stokes number is defined as a ratio between the characteristic timescale of the particle and that of the flow. Particles with higher particle timescales than fluid timescales experience less loss of velocity during a collision. The relation for the Stokes number used in the solver based on the inertia of the particle, including the added mass of the water, is given in equation 3.11.

$$St_{AM} = \frac{(\frac{\rho_s}{\rho_c} + C_{AM}) \, |\vec{u}_{n,in}| \, d_p}{9 \, \nu} \tag{3.11}$$

The relation of the restitution ratio for particles in fluids was found by Legendre [17]. First, Joseph et al. [13] measured the wet and dry restitution coefficients for particles of different materials and sizes. Legendre created expression 3.12 through all this data, which compares the wet restitution coefficient, normalized by its dry restitution coefficient, to the Stokes number of the particles. For the value of coefficient β, he proposed a value of 35.

$$\epsilon_{wet} = \epsilon_{dry} \, exp\left(-\frac{\beta_L}{St_{AM}}\right) \tag{3.12}$$

3.2.5 Changed viscous drag

The earlier described Cd formulation (equation 3.5), as used in the solver is a factor calibrated on particles in a water column where the fluid can freely flow around all sides of the particle. Particles close to a wall, as in the IFP, will have part of that flow restricted by the wall. Chhabra, Kumar, and Prasad [12] performed multiple tests with different sizes of spherical particles in a round glass tube ($0.114 \leq d_p/D \leq 0.58$). Multiple liquids were used to get a wide range of particle Reynolds numbers ($10^{-6} \leq Re_s \leq 3000$).

They found that particles with the higher Reynolds numbers ($250 \leq Re_s \leq 3000$), experience a drag force with a C_d value 3 times as high as from equation 3.5. It is important to note that the angles of the pipe used in the research of Chhabra, Kumar, and Prasad were only up to 30 degrees, and particle velocities have been used up to a particle Reynolds number of 3000. The angles of the IFP almost all exceed 30 degrees, and the settling velocities between 0.4 and 1.2 m/s give particle Reynolds numbers of 5600 to 16800. Due to these differences, the solver does not use the C_d values of Chhabra, Kumar, and Prasad. The factor of 3 is used in Appendix D as a reference value to compare with the factor found from experimental data.

3.2.6 Wall effect

Particles that settle in vertical pipes experience a so-called wall effect. Due to the presence of the walls, particles settle less fast than they would without the walls. This effect is stronger for cases with a larger particle-pipe diameter ratio. The reason for this effect is that inside a pipe, the volume of the settling particle needs to be compensated by a flow over the limited cross area, creating a flow up and slowing the particle down. Without the walls, this same volume would be compensated over an infinite area, and the resultant flow would be negligible. This effect is included in the used solver by the way the undisturbed flow velocity is determined. The effect gives a settling velocity reduction of 1.5% for the particles of 14mm in the 120mm tube.

3.3 Physics of particle groups

3.3.1 Particle-particle interaction

Particles interacting with each other will experience the same normal and tangential forces as they do when interacting with a wall. These forces are included in the used solver by the soft-sphere model; more information about this model can be found in the dissertation of Nieuwboer [27]. According to Yang and Hunt [20], normal inter-particle collisions behave the same as particle-wall collisions.

3.3.2 Hindered settling

Particle groups do not reach the same terminal velocity as a single particle. This effect was first described by Richardson and Zaki [1] and a relation was given for the reduced settling velocity. Winterwerp [11] identified that the reduction of settling velocity is caused by three separate effects. The first effect is that a large number of particles moving down creates a return flow through the entire field, which slows the particles down. The solver includes this effect by how the undisturbed fluid velocity is found.
The second effect is that a mixture containing water and rocks has a higher average density than just water. Resulting in a higher static pressure and so in a higher buoyancy force. The solver includes this effect by using the concentrations for the buoyancy calculations.
The third effect is the increased fluid flow resistance due to the presence of other particles. This can be described as a sort of higher viscosity. The solver does include this by introducing a factor to the drag force depending on the fluid concentration and a factor β as shown in equation 3.13. The used factor for β is the one described by Di Felice [7] and is given in equation 3.14.

$$\vec{F}_d = \vec{F}_{d,0}\ \alpha_c^{-\beta+1} \tag{3.13}$$

$$\beta = 3.7 - 0.65\ exp(-\frac{(1.5 - log_{10}Re_p)^2}{2}) \tag{3.14}$$

With:
$\vec{F_d}$ as the drag force on a particle influenced by the presence of neighboring particles [N]
$\vec{F_{d,0}}$ as the drag force on a single particle in the absence of surrounding particles [N]
α_c as the fluid volume fraction [-]
β as the hindered settling exponent for the drag force [-]
Re_p as the particle Reynolds number depending on just the particle velocity. [-]

The particle Reynolds number Re_p used in equation 3.14 for the hindered settling exponent differs from the particle Reynolds number (Re_s) used for all the other applications. Compared to Re_s, the Re_p does not use the slip velocity (difference in particle velocity and fluid velocity), but only the particle velocity. Equation 3.15 gives the formulation used in the solver of Nieuwboer [27].

$$Re_p = \frac{|\vec{u}_p| d_p}{\nu} = \frac{\alpha_c \, |\vec{u}_{c|p} - \vec{u}_p| d_p}{\nu} = \alpha_c \, Re_s \tag{3.15}$$

3.4 Physics of non-round particles

3.4.1 Drag force

The drag formulation in equation 3.5 and the wall correction discussed in section 3.2.5 are calibrated for spheres. The rocks placed using an IFP and during Fontijn's [30] model lab tests are not spherical. The difference in shape influences the C_d value of the particle. Two sources for the difference in C_d value due to the shape are investigated. The values from these sources are only used as reference values and not used directly since the C_d value is dependent on the rock shapes and the shape properties of the rocks used in the experiments of Fontijn [30] are unknown.

The first source is Van der Wal [15], which gives the mean and standard deviation values for the C_d values found in a large set of quarry rocks.

- Mean $C_d = 1.42$ [-]
- Standard deviation of $C_d = 0.33$ [-]

The second source is from Wu and Wang [19], who give an equation for the C_d value depending on the Corey shape factor S_f. The equation was found by analyzing rock, sand, and sediment data from multiple experiments in literature. The Corey shape factor is defined as $S_f = c/\sqrt{a\,b}$ in which a, b, and c equal the lengths of the longest, intermediate, and shortest axes of the particle. In table 3.2 C_d values for different shape factors are given.

Table 3.2: C_d values according to Wu and Wang (2006) for different shape factors of the particle and the factor by which the rock C_d value is larger than that of equation 3.5.

	$S_f, 0.5$	$S_f, 0.7$	$S_f, 0.9$	$S_f, 1$
C_d rock ($Re_s, 10^3$)	1.66	1.02	0.62	0.48
C_d rock ($Re_s, 10^4$)	1.62	0.98	0.60	0.46
C_d rock/C_d sphere	3.52	2.14	1.30	1.01

3.4.2 Lift forces

The asymmetric shape of rocks induces asymmetric flow fields around them. These flow fields can cause a lift force perpendicular to the flow direction. The general formulation for the lift force is given in equation 3.16.

$$\vec{F_L} = \frac{1}{2} \, C_L \, A_p \, \rho_c \, (\vec{u}_{c|p} - \vec{u}_p) \, |\vec{u}_{c|p} - \vec{u}_p| \tag{3.16}$$

With C_L as the lift coefficient.

This C_L is dependent on the shape of the rock, the orientation and its Reynolds number. These forces are not included in the used solver. As a result, the particles will experience less force in the direction perpendicular to the flow.

4 CFD-DEM model

The numerical model used in this thesis is the semi-resolved CFD-DEM solver of Nieuwboer [27]. The solver combines the methods of finite volume computational fluid dynamics (CFD) for the fluid with the discrete element model (DEM) for the particles. This chapter provides an explanation of the finite volume CFD and DEM models, as well as the semi-resolved coupling between both. A more detailed description of the solver can be found in the work of Nieuwboer [27].

4.1 Finite Volume CFD

Fluid motion is governed by the Navier-Stokes equations, which describe the conservation of momentum of a fluid. In finite-volume computational fluid dynamics, the domain is divided into small control volumes. The Navier-Stokes equations are then solved for every control volume and time step to calculate the fluid movements. The fluid is assumed incompressible with a variable fluid volume fraction or continuous phase fraction (α_c). The fluid fraction is the part of the volume filled with fluid, whereas the other part is filled with pieces of rock.

Equation 4.1 gives the modified Navier-Stokes equation for the fluid fraction. The terms on the left-hand side of the equation represent the fluid's inertia and the momentum's advection. The first term on the right-hand side is the pressure gradient. This is most often, together with gravity, the driving force for fluid movement. The second right-hand term represents the stresses that are applied to the fluid. These stresses are caused by the fluid viscosity and the turbulent stresses and will act as a diffusive term. The last two terms are the gravity and the interaction force per unit volume of the solid particles acting on the fluid. In the case of the IFP, the interaction forces will act as the main driving force for the fluid.

$$\frac{\partial\, \alpha_c\, \rho_c\, \vec{u}_c}{\partial\, t} + \nabla \cdot\ (\alpha_c\, \rho_c\, \vec{u}_c\, \vec{u}_c) = -\nabla p + \nabla \cdot (\alpha_c\, \bar{\bar{\tau}}) + \rho_c\, \vec{g} + \vec{f}_i \tag{4.1}$$

With:
α_c as the continuous phase fraction [-]
ρ_c as the continuous phase density [kg/m^3]
$\vec{u}_c$ as the continuous phase velocity [m/s]
t as the time [s]
p as the pressure acting on the continuous phase [N/m^2]
$\bar{\bar{\tau}}$ as the effective stress tensor, including both the viscous and turbulent shear stresses acting on the fluid [N/m^2]
$\vec{g}$ as the gravitational acceleration [m/s^2]
$\vec{f}_i$ as the interaction force of the discrete particles acting on the fluid per unit volume [N/m^3]

The interaction forces between the particles and the water used in the solver are defined in equation 4.2. These forces include the following: drag, added mass, pressure gradient force, viscous force, gravity, and buoyancy. An explanation of each of these forces can be found in chapter 3.

$$\vec{F}_i = -(\vec{F}_d + \vec{F}_{AM} + \vec{F}_{pg} + \vec{F}_{visc} + \vec{F}_g + \vec{F}_b) \tag{4.2}$$

With:
$\vec{F}_i$ is the interaction force between the fluid and the discrete particles [N]
$\vec{F}_d$ as the drag force on the particle [N]
$\vec{F}_{AM}$ as the added mass force on the particle [N]
$\vec{F}_{pg}$ as the pressure gradient force [N]
$\vec{F}_{visc}$ as the viscous force acting on the surface of the particle [N]
$\vec{F}_g$ as the gravitational force [N]
$\vec{F}_b$ as the buoyancy force [N]

The interaction force in equation 4.1 relates to this interaction force $\vec{F}_i$ as:

$$\vec{f}_i = \frac{\vec{F}_i}{V} \tag{4.3}$$

With:
V as a reference volume, typically the volume of a mesh cell [m^3]

4.2 DEM

The particle's movements are calculated using the Discrete Element Method. The method tracks the center of a particle throughout a domain by applying Newton's second law on the particle. For every time step, first, the particle's inertia is calculated with the forces on the particle using equation 4.4. Then, the new velocity and particle coordinates are calculated from the particle inertia. Afterward, a check and, if necessary, a calculation are performed for collisions.

The left-hand side of equation 4.4, is the particle inertia. On the right-hand side, the forces on the particle are found. These forces are coupling forces between the particle and the fluid and are the same as in 4.2.

$$V_p \rho_p \frac{d\vec{u}_p}{dt} = \vec{F}_d + \vec{F}_{AM} + \vec{F}_{pg} + \vec{F}_{visc} + \vec{F}_g + \vec{F}_b \tag{4.4}$$

With:
V_p as the volume of the particle [m^3]
ρ_p as the density of the particle [kg/m^3]
$\vec{u}_p$ as the particle velocity [m/s]
t as the time [s]

4.3 Semi-resolved coupling

The coupling terms between the continuous fluid and the discrete particles are already included in the equations of section 4.1 and 4.2. The forces in these terms follow from the physics captured with the model. The term semi-resolved refers to the way the fluid is coupled to the particles, how the data for the forces is gathered from the grid for each discrete point particle, and how the data from the point particles is projected to the grid.

Semi-resolved coupling is used for grid size and particle size ratios of 1/10 to 3, as is visualized in figure 4.1 from Ma et al. [26]. The working principle is explained in the following sections of this chapter. When a particle is larger than 10 times the grid size, the pressures per cell can be integrated over the particle boundary to gain the total force on the particle. This coupling method is referred to as resolved CFD-DEM. When particles are smaller than 1/3th of the grid size, only the velocity of the grid cell that the particle is occupying affects the particle, and thus, this one is used for the coupling. This case is referred to as unresolved CFD-DEM.

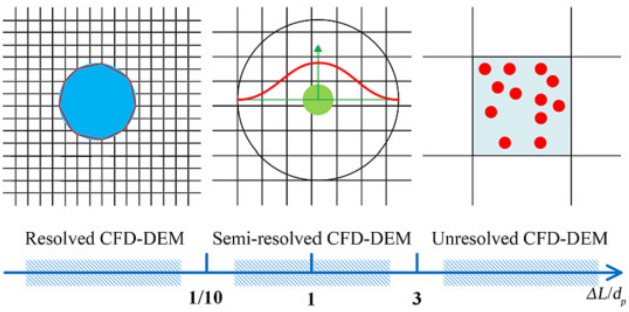


Figure 4.1: A visualization of the grid size particle size ratios for resolved, semi-resolved and unresolved CFD-DEM. [26]

4.3.1 Mapping particle information to the finite volume mesh

The solver uses a Kernel function as described by Xiao and Sun [22] for mapping the particle information to the finite mesh volume. The concentration of a single particle is distributed over multiple cells by using the Gaussian kernel from equation 4.5.

$$\phi(r_{j,n}) = \frac{1}{(\sigma\sqrt{2\pi})^{n_d}} exp\left(-\frac{r_{j,n}^2}{2\sigma^2}\right) \qquad \text{with} \qquad r_{j,n} = |\vec{c}_j - \vec{c}_{p,n}| \tag{4.5}$$

With:
$\phi(r_{j,n})$ is the kernel for mapping the particle information of particle n to cell n of the Eulerian mesh $[m^{-3}]$ (for a 3-dimensional simulation)
$r_{j,n}$ is the distance from the current cell j to the center of particle n [m]
n_d is the number of spatial dimensions of the simulation. Since the integral of the function should be 1, the peak of the function decreases when computing it for a higher dimension [-]
σ is the standard deviation of the Gaussian kernel [m]
$\vec{c}_j$ is the location of the cell center of cell j [m]
$\vec{c}_{p,n}$ is the center of particle n [m]

The particle fraction in cell j is computed by the kernel function (equation 4.5) multiplied by the cell volume and the fraction of the particle volume over cell volume:

$$\alpha_{p,j} = \sum_{n=0}^{N_p} \phi(r_{j,n}) V_{cell,j} \frac{V_{p,n}}{V_{cell,j}} \tag{4.6}$$

To increase the efficiency of the method, 2 simplifications are applied. The first one is to limit the domain for which the kernel is evaluated. The domain within three standard deviations contains about 99% of its volume, as depicted in equation 4.7, and will only lead to a minor error.

$$\int_0^{\infty} \phi(r)dr = 1 \qquad\qquad \int_0^{3\sigma} \phi(r)dr \approx 1 \tag{4.7}$$

The second simplification is to approximate the exponent function from equation 4.5, which is expensive to solve, with the power function of equation 4.8 as also used by Nieuwboer [27]. This power function is smooth and approximates the function well within the domain of $r \leq 3\sigma$.

$$exp\left(-\frac{r^2}{2\sigma^2}\right) \approx \begin{cases} \left(1 - \left(\frac{r}{3\sigma}\right)^2\right)^4 & \text{if} \quad \left(\frac{r}{3\sigma}\right)^2 \leq 1.0 \\ 0 & \text{if} \quad \left(\frac{r}{3\sigma}\right)^2 > 1.0 \end{cases} \tag{4.8}$$

The interaction forces from the particle on the fluid use the same kernel function as described above for the concentration α_p. Just with the particle volume substituted for the interaction force, as shown in equation 4.9.

$$f_j = \sum_{n=0}^{N_p} \phi(r_{j,n}) V_{cell,j} \frac{F_{i,n}}{V_{cell,j}} \tag{4.9}$$

With:
$\vec{f}_j$ as the interaction force per volume for cell j $[N/m^3]$, $[kg/m^2s^2]$
$\vec{F}_{i,n}$ as the interaction force between the fluid and the particle n [N]

4.3.2 Mapping fluid properties to the particle

A different kernel function is used for mapping the fluid velocity to the particle than for mapping the concentration to the mesh. The undisturbed fluid velocity is required for the calculation of the drag force. As a result of applying the interaction force to the fluid phase, the fluid accelerates with the particle. Using the same kernel for calculating the fluid velocity would result in the disturbed fluid velocity. A quadratic distance weighing function is used to predict the undisturbed fluid velocity. This weighing function has a zero weight factor for the fluid velocity at the center of the particle and a larger one for the fluid velocity

further away from the particle. The limit of the domain (R) is just as for the other kernel function three times the normal distribution (3σ).

When taking an integral over the volume of this quadratic function, one will find a volume of $1/2\pi R^2$ for 2D cases and a volume of $4/5\pi R^5$ for 3D cases. The scaled quadratic distance weighing functions for 2D and 3D cases are given in equation 4.10.

$$\begin{aligned} \psi_{2D}(r_{j,n}) &= \begin{cases} \frac{2}{\pi R^4} r_{j,n}^2 & \text{for} \quad \left(\frac{r_{j,n}}{3\sigma}\right) < 1.0 \\ 0 & \text{for} \quad \left(\frac{r_{j,n}}{3\sigma}\right) \geq 1.0 \end{cases} \\ \psi_{3D}(r_{j,n}) &= \begin{cases} \frac{5}{4\pi R^4} r_{j,n}^2 & \text{for} \quad \left(\frac{r_{j,n}}{3\sigma}\right) < 1.0 \\ 0 & \text{for} \quad \left(\frac{r_{j,n}}{3\sigma}\right) \geq 1.0 \end{cases} \end{aligned} \tag{4.10}$$

With: $\psi_{2D(rj,n)}$ as the kernel function in a 2D case for mapping the Eulerian information of cell j to the particle n $[m^{-2}]$
$\psi_{3D(rj,n)}$ as the kernel function in a 3D case for mapping the Eulerian information of cell j to the particle n $[m^{-3}]$

Using the kernel functions, the velocity and fluid concentration at the particle can be calculated with equations 4.11 and 4.12.

$$\vec{u}_{c|p,n} = \sum_{n=0}^{N_{cells}} \psi(r_{j,n}) V_{cell,j} \vec{u}_{c,j} \tag{4.11}$$

$$\alpha_{c|p,n} = \sum_{n=0}^{N_{cells}} \psi(r_{j,n}) V_{cell,j} \alpha_{c,j} \tag{4.12}$$

With: $\vec{u}_{c|p,n}$ is the undisturbed fluid velocity for particle n $[m/s]$
$\vec{u}_{c,j}$ is the fluid velocity at cell j $[m/s]$
$\psi(r_{j,n})$ is the kernel for mapping the Eulerian information of cell j to the particle n computed with either $\psi_{2D}(r)$ or $\psi_{3D}(r)$ $[m^{-3}]$
$r_{j,n}$ is the distance between the center of cell j and the center of particle n [m]
$\alpha_{c|p,n}$, is the fluid fraction at particle n [-] $\alpha_{c,j}$ is the fluid fraction at cell j [-]

4.3.3 Distance factor

Note that for both kernels mentioned in the previous sections, the width or limit of the domain is three times the standard deviation. This standard deviation is chosen for each simulation by setting a distance factor γ, which relates the particle diameter (d_p) to the standard deviation (σ) according to equation 4.13.

$$\sigma = \gamma \, d_p \tag{4.13}$$

Finding the correct distance factor is important for the simulations of the IFP. For too small values, the particles will not get smoothed enough over the cells. Too large values and velocities far from the particle will be used and averaged for the undisturbed particle velocity, which creates issues with large shear flows like in the IFP.

5 Methodology

In this chapter, the methodology adopted for answering the research questions is discussed. The chapter consists of six sections.

The first section discusses the software used for the simulations: both the OpenFOAM solver of Nieuwboer and the Apptainer container software.
The second section shows the steps taken to verify whether the solver works as intended.
The third section covers the different parameters and the grid design of the solver, the values that have been chosen for these parameters during the simulations, and for which of these parameters the influence on the simulations is researched further.
The fourth section shows the changes made to the solver to incorporate changes in the C_d value and how this C_d value is determined.
The fifth section discusses the setup of the simulations and how these will be compared to the IFP model data.
The sixth section will give an overview of the limitations of the way the tests are performed.

5.1 Software

5.1.1 OpenFOAM and Apptainer

The solver developed by Nieuwboer was created for OpenFOAM version v1712, an open-source Computational Fluid Dynamics software designed in Linux. As of the time of writing, this version of OpenFOAM is 7 years old. The C++ compilers needed for this OpenFOAM version limit its use to the Linux software: Ubuntu long-term support version 18.04 or older. Neither the computers available for this research nor the university computing cluster, DelftBlue, operate on this older Linux version.

To run the software on the newer Linux versions, a container software named Apptainer is used. This software allows for creating a container holding the Linux software Ubuntu 18.04, OpenFOAM version v1712, and Nieuwboer's solver. The entire Linux computer system is virtualized using this container, enabling simulations to run on any system with the Apptainer software installed. Multiple verification tests were performed to ensure that the solver works correctly after the container's installation, which is discussed in the following section, section 5.2.

5.1.2 Paraview

The data created by OpenFOAM and the solver is raw data divided over many text files organized per time step. From this data the required data needs to be collected. In this section, the software used for this process, Paraview, and the general procedures will be described. The procedures and used filters exclusively for certain cases will be explained per case in section 5.5.

The software used to process the raw data is Paraview. Paraview is an open-source post-processing software and is the default software for processing OpenFOAM data. It can both visualize the data and process it with its many filters. Before opening Paraview, the OpenFOAM "foamToVTK" function, with the option "-noZero", is used. This converts the OpenFOAM's output format to VTK-files, the standard file format of Paraview. Normally, OpenFOAM's output files can be read by Paraview. However, Paraview does not register the Lagrangian data in OpenFOAM format. The "-noZero" option is used to skip the 0 time step, as this time step does not contain data for all variables and gives errors on loading in Paraview.

5.2 Solver verification

The BigGaussDPMFoam solver of Nieuwboer [27] was created for OpenFOAM version v1712 on Ubuntu 18.04. Since these software versions are outdated, a software container was created using Apptainer. To verify that the solver is still working as designed, multiple physics verification steps of Nieuwboer's research are repeated. During these verifications, an issue was found, which was solved afterward.
Four functionalities of the solver were tested by small simulations. The first set of verification simulations is of a free-falling ball reaching terminal settling velocity. The second set of verification simulations is of a particle colliding with a wall. The third set is of two particles colliding with each other. The fourth set is of large amounts of particles settling and experiencing the hindered settling effect.

5.2.1 Terminal settling velocity

The first verification test simulates a free-falling particle reaching terminal settling velocity. During this simulation, a single particle starts in a free water column. The particle accelerates due to the difference in gravity ($\vec{F}_g$) and buoyancy ($\vec{F}_b$). When the particle starts moving, the added mass force ($\vec{F}_{AM}$) and drag force ($\vec{F}_d$) become non-zero. The drag force becomes larger as the particle accelerates until an equilibrium is reached. At this equilibrium, the particle has stopped accelerating. Thus, the added mass force is zero again. The relation $\vec{F}_g + \vec{F}_b + \vec{F}_d = 0$ can be used to calculate this terminal settling velocity. Multiple tests were performed with the parameters in table 5.1. The velocity over time is compared to a numerically calculated solution in Python in figure 5.1. The equation solved in Python is $V_p \rho_p \frac{\partial u_p}{\partial t} = \vec{F}_g + \vec{F}_b + \vec{F}_{AM} + \vec{F}_d$, with a time step of $10^{-7} s$. As can be seen, the results match well.

Table 5.1: Parameters for the terminal settling velocity verification tests.

Parameter	Quantity
Particle	Rock
d_p	0.01 m
ρ_p	2650 kg/m^3
Fluid	Water at 20 °C
ρ_c	998.2 kg/m^3
ν	$1.0035 \cdot 10^{-6}$ m^2/s
Domain	
Width	0.3 m
Δx	0.02, 0.01, 0.0067, 0.005, 0.002 m
$Dist.fac.\gamma$	1.0
Time	
Δt	$1 \cdot 10^{-3} s$
$N_{timeSteps}$	12

In the visual inspections of these simulations, a strange phenomenon was seen in the flowfield. The flow moved through the particle instead of around the particles. This was due to an error within the solver code; the interaction force per unit of volume $\vec{f}_i$ mentioned in equation 4.1, which should be exerted on the fluid, was not being calculated. This was caused by an incorrectly placed comment in the C++ code. Appendix B shows the change made to the solver code. The difference between the flow with the issue and the flow after correcting the issue can be seen in figure 5.2.

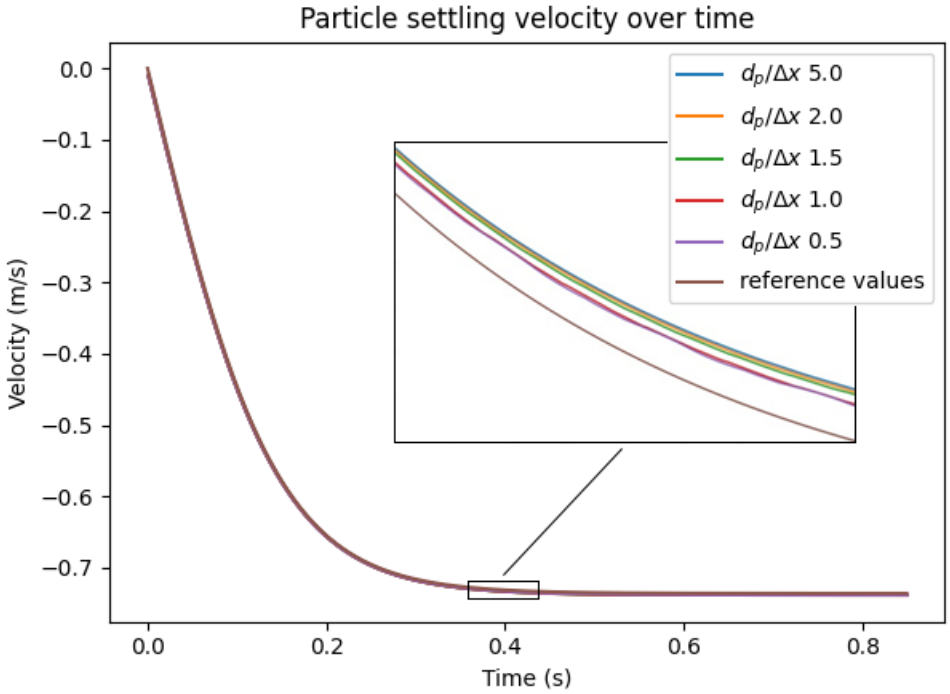
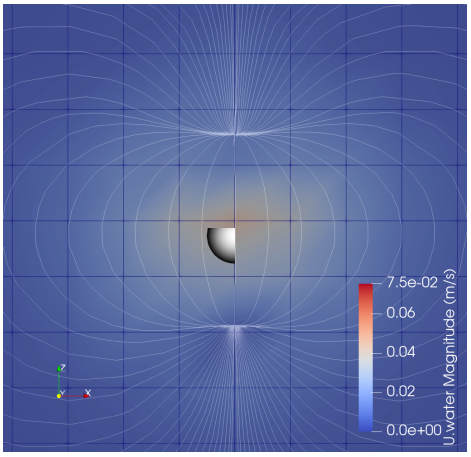
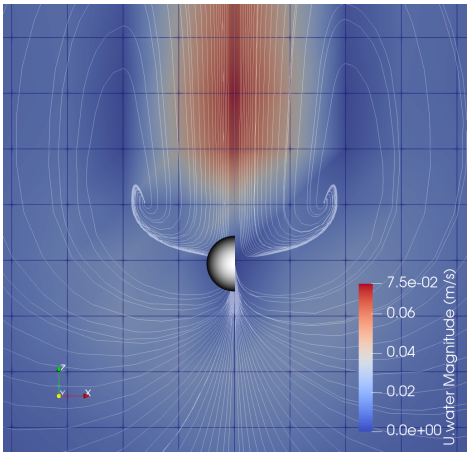


Figure 5.1: The single particle settling velocity over time for different grid sizes compared to the reference values from the in Python solved equation.



(a) The flowfield around (and through) a particle without the interaction force.



(b) The flowfield around a particle with the interaction force.

Figure 5.2: The difference in flow fields before (a) and after (b) solving the issue which caused the interaction force from the particle to the fluid to be absent.

5.2.2 Particle-wall collisions

The second set of tests are verification tests of particle-wall collisions. The first simulation is a particle with the same parameters as used by Nieuwboer as given in table 5.2. The simulation is of a metal ball in oil colliding with the wall based on the experimental results of Gondret et al. [14]. The results of the simulations are shown in figure 5.3. No direct differences are visible from the simulations performed by Nieuwboer [27].

Table 5.2: Parameters for the Gondret particle-wall collision tests.

Parameter	Quantity
Particle	Steel
d_p	0.003 m
ρ_p	7800 kg/m^3
E	240 Gpa
e_{dry}	0.97
$\lvert\vec{u}_{n,in}\rvert$	0.58 m/s
Re_s	164
St	152
Fluid	Silicon oil °C
ρ_c	935 kg/m^3
ν	$1.06952 \cdot 10^{-5}$ m^2/s
Domain	
Width	0.3 m
Δx	0.02, 0.01, 0.0067, 0.005, 0.002 m
Dist. fac. γ	3.0
Time	
Δt	$1 \cdot 10^{-3}$ s
$N_{timeSteps}$	12

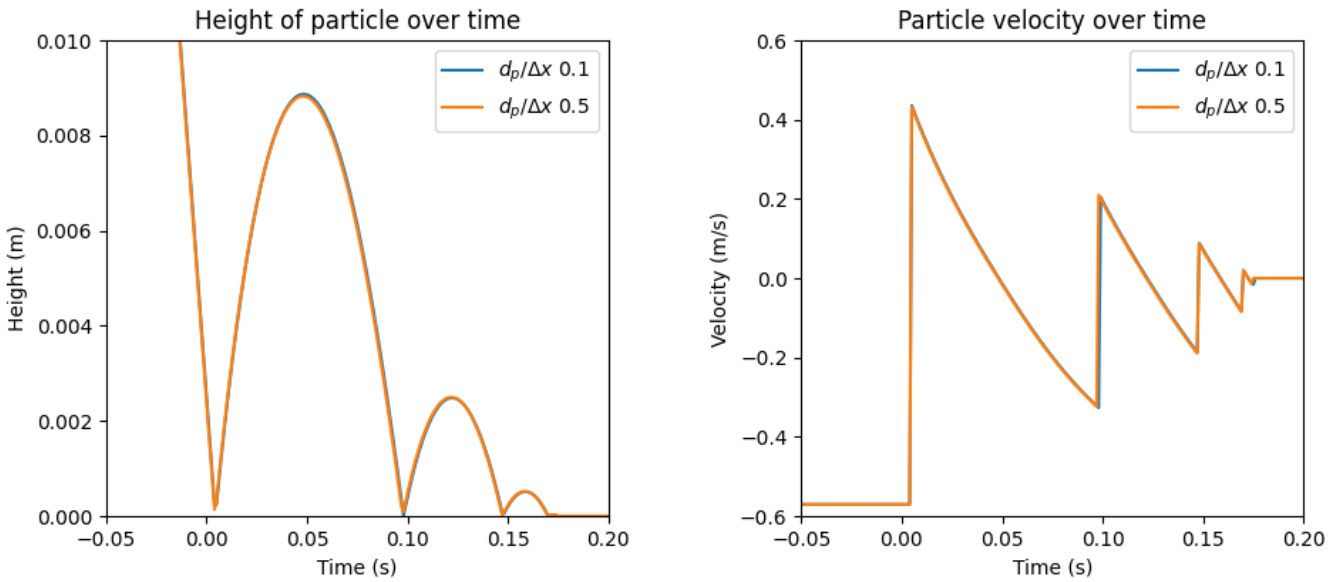


Figure 5.3: The height and velocity of a simulated particle colliding with a wall over time.

The second set of simulations is based on the restitution coefficients of particles in fluid. Four simulations were performed and compared to the Legendre relation (equation 3.12) in figure 5.4. The parameters for the simulations can be found in table 5.3. These data points coincide with the expression of Legendre.

Table 5.3: Parameters of the Legendre particle-wall collision simulations.

Parameter	Quantity
Particle	
d_p	0.0005, 0.001, 0.003, 0.030 m
$\lvert\vec{u}_{n,in}\rvert$	$0.069, 0.18, 0.57, 2.61\ m/s$
ρ_p	$7800\ kg/m^3$
E	$60\ Gpa$
ϵ_{dry}	0.97
St	$3, 16, 152, 2455$
Fluid	Silicon oil
ρ_c	$935\ kg/m^3$
ν	$1.06952 \cdot 10^{-5}\ m^2/s$
Time	
$N_{timeSteps}$	12

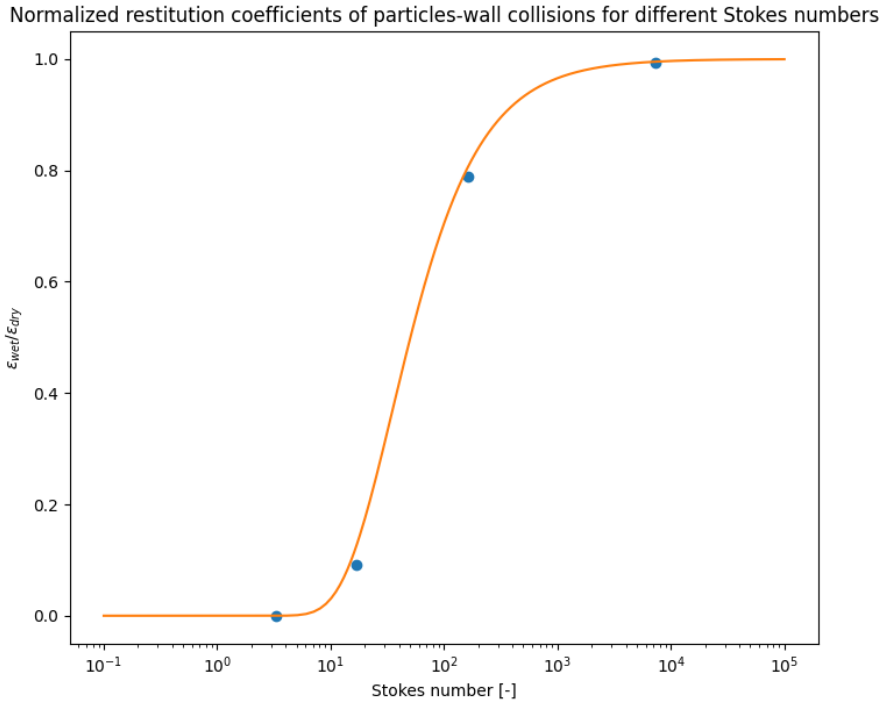


Figure 5.4: Normalized restitution coefficients over Stokes number of four particle-wall collision simulations compared to the relation of Legendre (equation 3.12).

5.2.3 Colliding particles

The third verification test is like the second test, a comparison between the restitution coefficient and the trendline of Legendre. In these simulations, 2 particles collide with each other. Four simulations with different parameters are compared to the Legendre relation (equation 3.12). The parameters used for the tests can be found in 5.4. The results are plotted together with the Legendre relation in figure 5.5. The data points lay close to the Legendre relation, and the deviation can be attributed to some uncertainties during the tests.

Table 5.4: Parameters of the particle-particle collision simulations.

Parameter	Quantity
Particle	
d_p	0.001, 0.001, 0.003, 0.030 m
$\lvert\vec{u}_{n,in}\rvert$	$0.049, 0.22, 0.64, 2.85\ m/s$
ρ_p	$7800\ kg/m^3$
E	$60\ Gpa$
ϵ_{dry}	0.97
St	$3, 16, 152, 2455$
Fluid	silicon oil
ρ_c	$935\ kg/m^3$
ν	$1.06952 \cdot 10^{-5}\ m^2/s$
Time	
$N_{timeSteps}$	12

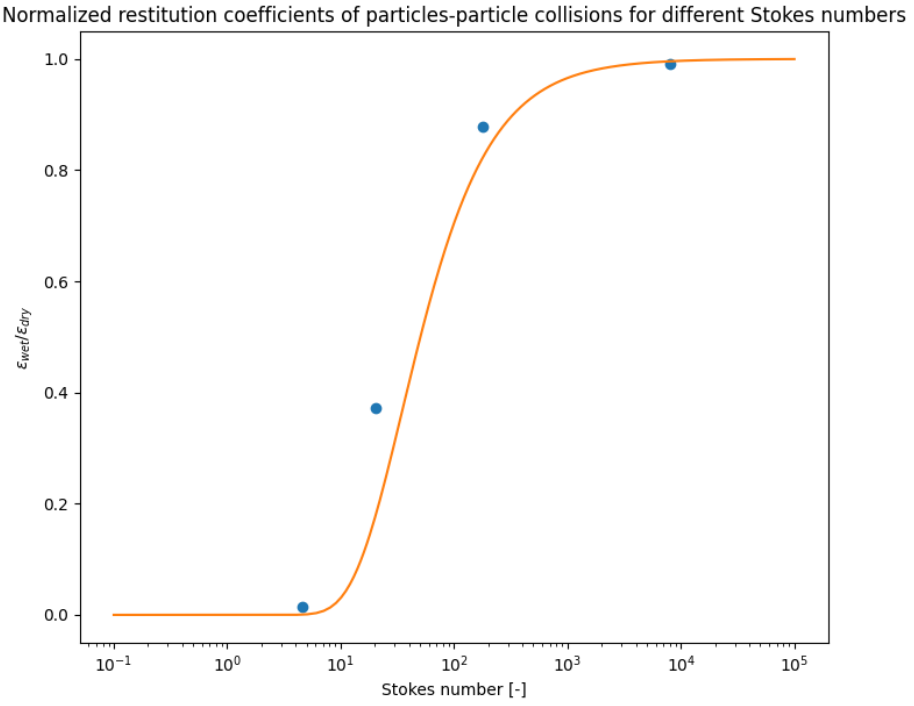


Figure 5.5: Normalized restitution coefficients over Stokes number of four particle-particle collision simulations compared to the relation of Legendre (equation 3.12).

5.2.4 Hindered Settling Effect

Particles close to each other experience a hindrance from the other particles around them while settling. This so-called hindered settling effect is taken into account in the drag formulation by the factor given in equation 3.14. Multiple settling simulations are performed to verify this part of the code. Each has different starting solid concentrations. These concentrations should predict the correct settling velocities according to the hindered settling formulation. The parameters for the hindered settling simulations can be found in table 5.5.

The particles' average settling velocity is compared to a reference solution calculated in Python with time steps of 10^{-7}. Both the reference solution and the average settling velocity over time can be found in figure 5.6. It can be seen that the velocities averaged over the particles mostly follow the reference data. The data does deviate after some time. This can be attributed to the particle column slowly diluting over a larger volume and the concentration reducing in the simulations due to the particles at the bottom settling faster and the particles at the top settling slower. The lower concentration results in the higher settling velocities seen in figure 5.6.

Table 5.5: Parameters of hindered settling simulations.

Parameter	Quantity
Particle	
d_p	0.01 m
ρ_p	2650 kg/m^3
α_p	$0.027, 0.1, 0.21, 0.31$
Domain	
Δx	0.01 m
Fluid	water at 20 °C
ρ_c	998 kg/m^3
ν	$1.0034 \cdot 10^{-6}$ m^2/s
Time	
Δt	$2 \cdot 10^{-3}$ s
$N_{timeSteps}$	20

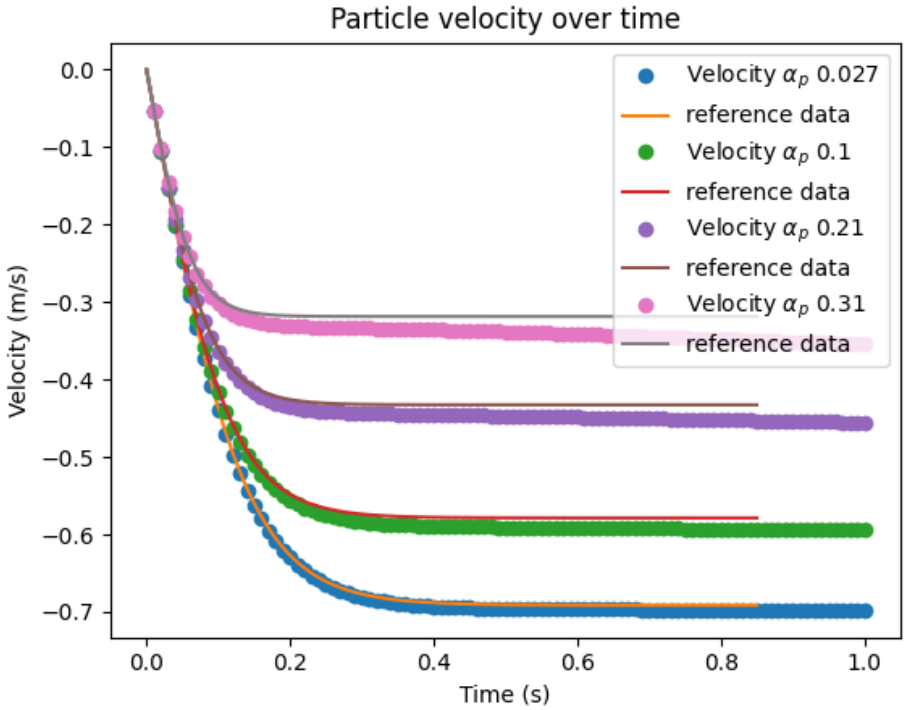


Figure 5.6: Hindered settling simulations compared to the reference data created in Python.

5.3 Solver parameters and grid design

The solver has many parameters that influence the outcome of the simulations when set incorrectly. Small test simulations determine which values can be used for certain parameters. Other parameters are physical properties chosen to represent the model lab tests as good as possible. Some parameters are coupled to the calculation grid used. Hence, the choices for the grid design are discussed in this chapter, too. First, the parameters for the fluid are discussed, then those of the grid, and last, the parameters for the particles.

5.3.1 Fluid parameters

The parameters for the fluid are the physical properties of the density of the fluid (ρ_c) and the kinematic fluid viscosity (ν). The values used for these parameters in the simulations with water are those generally known for fresh water at 20 $^\circ C$, $\rho_c = 998.2\ kg/m^3$, and $\nu = 1.0034 \cdot 10^{-6}\ m^2/s$.

5.3.2 Grid parameters and grid design

Parameters

The grid is created using OpenFOAM's generation software blockMesh and snappyHexMesh. BlockMesh is used to create a uniform, block-shaped grid, and snappyHexMesh is used to reshape this mesh to the desired shape of the domain. BlockMesh and snappyHexMesh together have a lot of parameters that can be set, but after generation, the size of the grid cells is the most important parameter for the grid. The other parameters are chosen to create a grid as homogeneous as possible over the entire domain. When the grid size is too large, the fluid calculations will contain large errors, while if the grid size is too small, the calculation times will become very large. The data from the verification tests in figure 5.1 shows that the particle velocity is not influenced by the grid cell size between $d_p/\Delta x$ values of 0.1 to 5. The final grid size will follow from the grid design steps. During the grid design steps and for the final test setup, a simulation with a finer grid is performed to check if the results stay the same, and thus, a small enough grid size is used.

2D or 3D

An often-used simplification in CFD simulations is to simulate a 2D variant of the problem instead of the entire 3D field. The difference in calculation time for the IFP simulations is about 1 hour for a 2D simulation compared to 12 hours for 3D simulations. Creating a 2D simulation of a 3D process is not always possible. The movement and differences in the flow field along one of the dimensions need to be negligible.
In this case, most of the flow is in a cylinder-formed tube. The first step would be to determine which dimension could be simplified. Since the particle is moving along the length of the pipe (z) and the flow shear is most prominent over the height of the pipe (x), the width of the pipe (y) is the only dimension that could be flattened/ taken constant. The cylinder shape creates different widths over the height of the pipe. The same process in a square pipe would be easier to simulate as a 2D object since the width would be the same over the height of the pipe.
When flattening the pipe, a choice needs to be made about what ratio to preserve: either the height of the particle to the height of the pipe or the cross-sectional area of the pipe covered with particles compared to the part covered with only fluid.

The cross-sectional area ratio should stay the same for the correct fluid velocity in the backflow. When applying this, the particle needs to shrink, or the pipe diameter has to become larger. The particle can not shrink because then the volume and thus the mass and gravitational force are not correct anymore. The pipe could be enlarged, but then the shear in the flow over the height is not correct anymore, in which case particle clusters would be able to become larger before being influenced by the backflow.
When the height ratio stays the same, the pipe's and particle's diameters stay the same. However, the backflow is incorrect.

The simulations with multiple particles will all be performed in 3D since 2D, in all cases, creates unfavorable changes to the particle or the fluid flow.

Domain and grid design

First, the shape of the domain will be discussed. The 3D domain consists of a pipe and a block-shaped flow field outside of the pipe. The pipe and the block-shaped flow field overlap partly, as shown in figure 5.7.

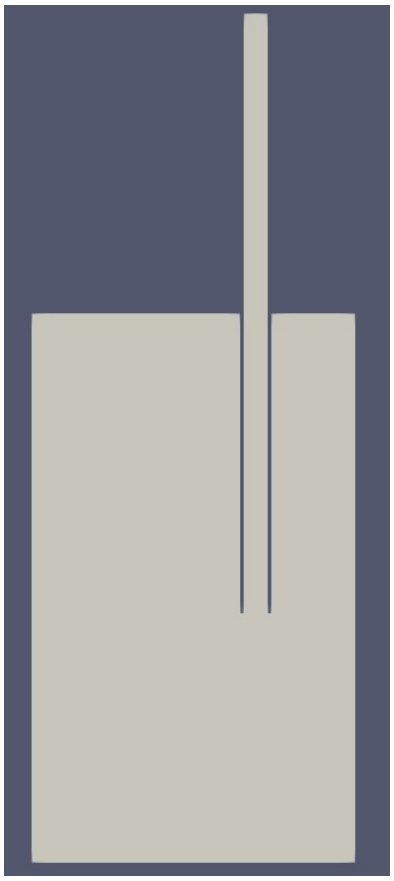
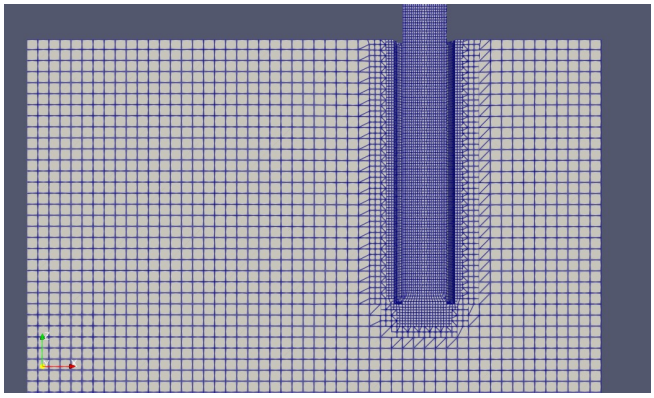
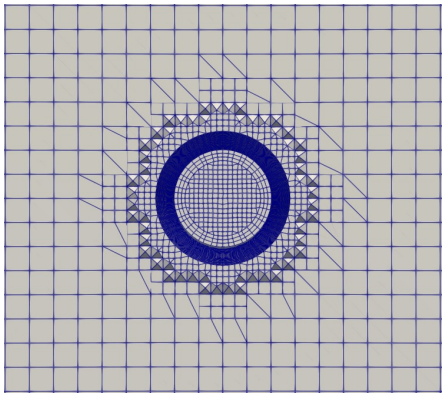


Figure 5.7: A cross-section of the domain, with the inside of a pipe and the flow field around it.

The pipe has the same diameter as in the lab model tests of 120 mm. The length of 2880 mm is about double the length used in the model lab tests, chosen so the upper half can be used as particle storage at the beginning of the simulation.
After performing test simulations with grids of different cell sizes, as shown in Appendix C, a cell size of about 7.5mm was chosen inside the IFP. The cells outside the IFP, with a size of 30mm, are a bit coarser, as the outside is less important to the simulations. The 3D grid consists of 312152 cells. Figure 5.8 shows the 3D grid used.



(a) The view from the side with the pipe sliced in half so the inside can be seen.



(b) A slice of the 3D grid along the cross-section of the pipe.

Figure 5.8: Two slices through the 3D grid showing the cell structure in and around the pipe.

5.3.3 Particle parameters

Rock size

The rock size is chosen to represent the rocks used in the model lab tests of Fontijn [30] as well as possible. The report used three different gradation sizes for the groups of rocks, which are shown in figure 5.9. These gradations need to be converted to representative particle diameters to use them in the CFD solver.

Figure 5.9: A sample of the three graduations used in Fontijn's lab tests. [30]

The three rock gradations used in the lab test are given as minimum and maximum values, these are: 8-12 mm, 12-16 mm, and 16-32 mm. The gradation values usually are the sizes of the sieves used by the supplier to sort the rocks into these gradations. As no other information is available, it will be assumed that the rock sizes within the gradation are normally distributed, and the median sieve diameter D_{50} equals the average of the gradation limits. In the simulations, the 12-16 mm gradation will be simulated. The rocks in the 16-32 mm gradation are not used as they have more movements influenced by their flatter shape, which the spheres in the solver can not predict. The 8-12 mm gradation rocks are not used since more of these rocks need to be simulated for the same productions, and more particles in simulations will cause slower simulations. The 12-16 mm gradation corresponds with the rock sizes in the real IFP of 0.16 to 0.24 meters.
Due to current solver limitations, all the particles in a single simulation will be spherical and have the same diameter and volume. The used stone size is chosen so that its volume equals that of a rock of the median sieve diameter. The corresponding diameter of a sphere D_{s50} is calculated using two equations from the rock manual [21] relating the nominal diameter (D_n) to the diameters mentioned before, equation 5.1 and equation 5.2. A note has to be made that the value 0.84 in equation 5.1 is not exact and deviates quite a bit depending on the sieving technique and stone properties.

$$D_{n50} = 0.84\, D_{50} \tag{5.1}$$

$$D_n = (\pi/6)^{1/3}\, D_s \tag{5.2}$$

The equations taken together in equation 5.3 give the relation between the median sieve diameter and the median equivalent sphere diameter and show these diameters do not deviate a lot from each other. Taking the large uncertainty in the form factor of equation 5.3, the values D_{50} and D_{s50} values will be considered equal to each other.

$$D_{s50} = \frac{0.84}{(\pi/6)^{1/3}}\, D_{50} \approx 1.04\, D_{50} \approx D_{50} \tag{5.3}$$

Distance factor γ

As discussed in section 4.3.3, the distance factor is the relation between the particle diameter and the standard deviation of the Gaussian kernel. This Gaussian kernel is used to spread the particle concentration over the grid cells and couple interaction forces from the particles to the fluid. The outer limit of three times the standard deviation is also used in the Gaussian kernel and in the weighted quadratic function used to couple

the fluid velocity to the particles.
This means that using incorrect values for the distance factor will cause incorrect particle volume concentrations in the grid cells and incorrect interaction forces between the particle and fluid both ways.
Too low values of the distance factor will limit the cells used to spread the particle volume. When only one or several cells fall within the kernel domains, it will cause discontinuous movement of the particle concentration over the grid as well as particle fluid velocities to be influenced by their interaction force.
In extreme cases, for a distance factor with a kernel domain that is smaller than the particle size and smaller grid cells than the particle size, all the volume is placed within a grid cell with less volume than necessary, which causes volume loss.

Too large distance factors also cause issues. With a high γ, the undisturbed fluid velocity is measured further from the particle. That is not an issue in fluids with uniform flows, but the process in the IFP consists of a mixture flow down and a fluid flow up, resulting in a high shear flow. This means the return flow will influence the average flow when taking the fluid velocity from too far away. Another reason the distance factor should not be taken too high is that the amount of cells that are checked per particle depends on the distance factor. The simulations will take much longer to run for larger distance factors.

Five simulations of single settling particles with distance factors from 0.3 to 3 were performed to get to know the lower limits of the distance factor. The grid cells Δx and the particle diameters d_p in these simulations are of equal sizes. The results of these are shown in figure 5.10. From the velocities, it can be seen that the distance factors of 0.3 and 0.5 are too small. From 0.8 and upwards, the particle velocity is calculated correctly.

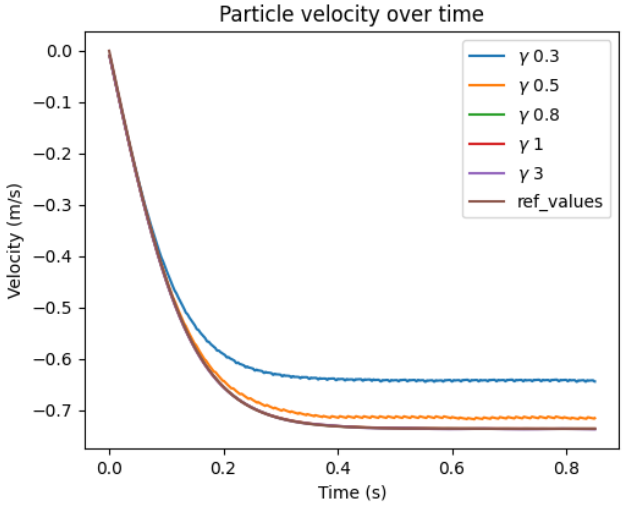


Figure 5.10: The particle velocity of a single settling stone. Simulated with different values for the distance factor γ.

The distance factor is only part of the reason the velocities are incorrect. The full issue is that the kernel domain includes not enough grid cells. The kernel size is given by $3\gamma d_p$, and the number of cells inside the domain depends on the cell size Δx. This means the full relation would be $\frac{\gamma\, d_p}{\Delta x} \geq 0.8$.
With the simulations using a particle size of 0.014 meters and grid cells between 0.0075 and 0.030 meters, the distance factor could be 0.4 inside the IFP and 1.6 outside the IFP. As the flow outside the IFP is less important, and Nieuwboer used the lowest value of 1, a value of 1 was chosen. A simulation with a different distance factor will be performed with the full simulation setup to see the influence of the distance factor in the larger simulations.

Particle starting positions

In the model lab tests, the rocks were introduced as a constant stream from a conveyor belt above the pipe. Since the simulation is entirely below the water surface and creating new particles during the simulation is not an option with this solver, the particles must be introduced at the beginning of the simulation.

Some setup tests were performed to study the placement of the particles. During these tests, 2 variables were looked into. The first variable was the location of particles, whether they should start all in the holding area

(upper half of the pipe) or along the full length of the pipe. The second variable is the distribution of the particles, either in an orderly grid or randomly distributed.

A comparison between a simulation with only particles in the upper half of the pipe and one with particles placed along the full length of the pipe can be seen in figure 5.11 and figure 5.12 . Both the simulations are shown after 0.5 seconds and 2.0 seconds. When comparing the cases, it can be seen that the flow field in the pipe only develops near the location of the particles. This means a simulation in which the pipe is entirely filled with particles earlier develops a full flow field; this can be seen at 0.5 seconds. At 2.0 seconds, a startup clustering phenomenon can be seen at the half-filled pipe. This phenomenon was also visible in the lab model tests, but since it is present in the first seconds of the placement process, this is not of interest to this research. In the half-filled pipe, there are only limited particles that are not influenced by this phenomenon. The fully filled pipe does not show this phenomenon and has many particles that can be studied.

Figure 5.11: The comparison of the pipe starting half-filled with particles (on the left) to the pipe fully filled with particles (on the right) at 0.5 seconds.

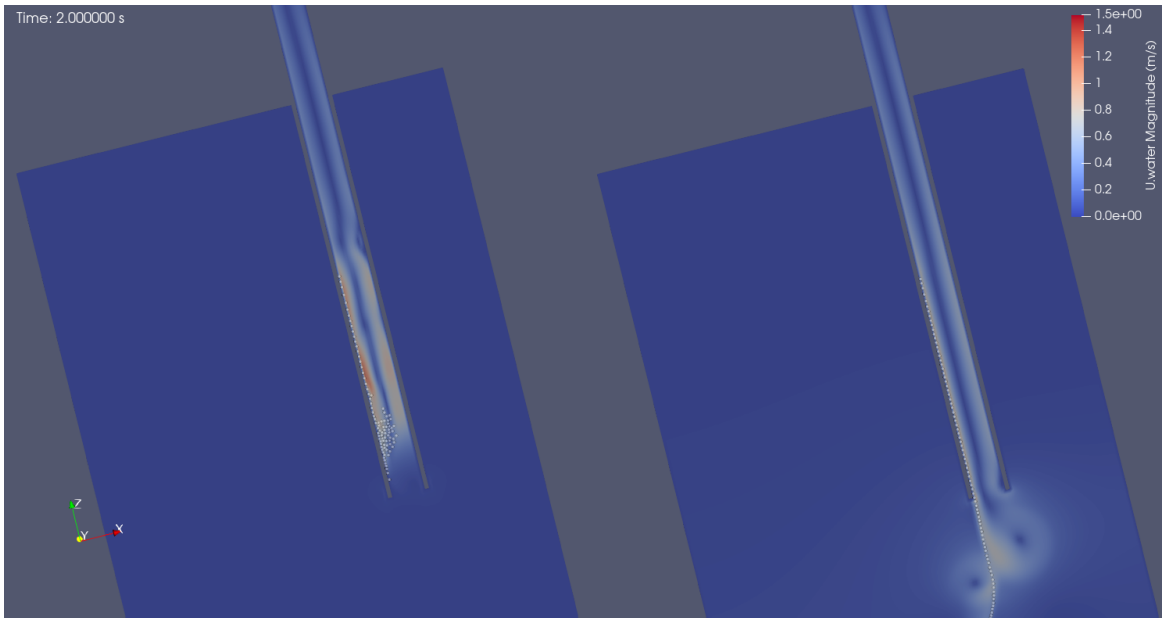


Figure 5.12: The comparison of the half-filled pipe with the fully filled pipe at 2 seconds.

In figure 5.13 and figure 5.14, the comparison between two pipes, one starting with evenly distributed particles and one with randomly placed particles, can be seen at 0 seconds and after 1 second of simulation. The figure shows what the particle locations and the flow field look like at 1 second after the start of the simulation. In the simulation in which the particles start placed randomly over the pipe, particle clusters can be seen to form, while the simulation starting with the evenly distributed particles shows that the particles stay in one straight line. The clustering behavior is also found in the model lab tests, showing the randomly placed rocks is the better option of the two.

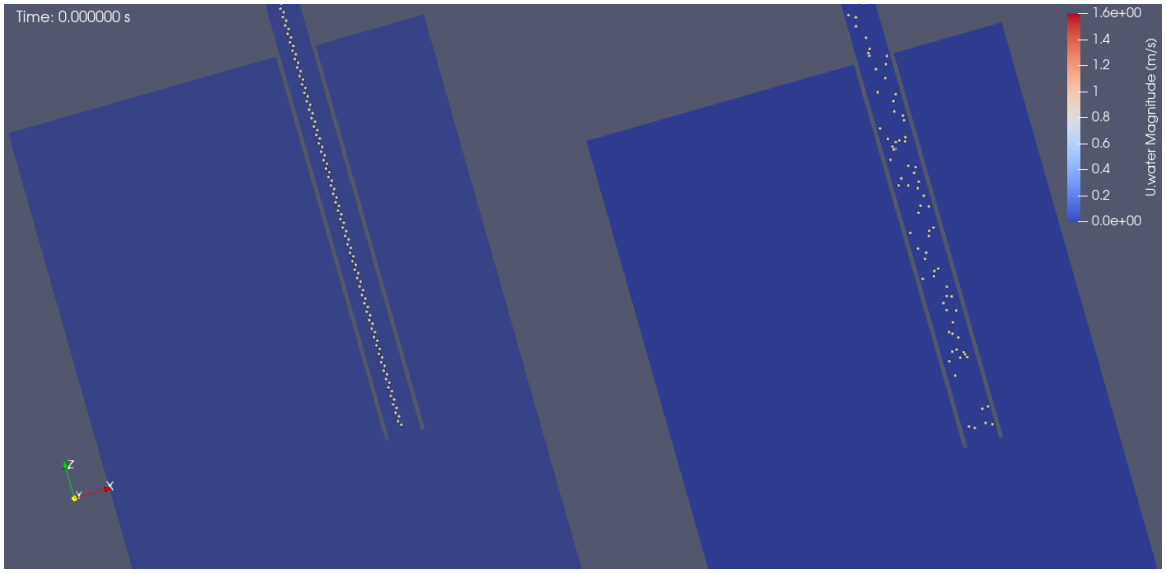


Figure 5.13: A comparison between simulations starting with evenly distributed particles (on the left) and randomly placed particles (on the right) at 0 seconds.

Figure 5.14: A comparison between simulations starting with evenly distributed particles (on the left) and randomly placed particles (on the right) after 1 second of simulation.

The simulations will start with particles randomly distributed over the entire pipe length. The particle location file with randomly distributed positions will be created with the use of Python. One file will be created per starting particle concentration. These files will be used for the simulations varying the angle and C_d value; this way, the particles are randomly distributed, but the simulations all have the same starting positions.

5.4 C_d value

According to the literature sources mentioned in both section 3.2.5 and section 3.4.1, particles moving along the side of a pipe and non-spherical particles experience higher drag force due to higher C_d values. For a particle rolling along a pipe wall, the $C_d$value could be about 3 times higher than a particle falling in the free water column, and for a non-round particle, the $C_d$value could be about 2 to 3 times higher than for a round particle.
The issue, however, is that both these numbers are uncertain. The value given by literature for the particle along the walls was found with particle Reynolds numbers smaller than those of the particles in the model IFP. The value for the non-spherical particles depends greatly on the shape of the particle, and as the shape factor of the used rocks is unknown, this value can not be given with certainty.

Fontijn has performed some tests with single rocks and one marble, all in the model IFP under an angle of 75 degrees. This data is used to find the factor by which the C_d value would need to be multiplied according to this situation. The found factor is 3. Appendix D contains more information about the steps taken to find this value.

Unfortunately, changing the C_d value in the solver is not that straightforward. The drag model can be chosen as part of the particle properties, but since the hindered settling factor (as discussed in section 3.3.2) is part of the diFeliceDrag model, this one must be used. The part of the C_d value uncorrected for the hindered settling is calculated using the Brown and Lawler formulation (equation 3.5). This can not be changed without changing part of the solver, written in C++. A parameter C_{factor} is added to keep the changes to a minimum, which is multiplied with the original C_d value as shown in equation 5.4. This parameter C_{factor} can be set as part of the particle properties.

$$\vec{F_d} = \frac{1}{2}\, C_{factor}\, C_d\, A_p\, \rho_c\, (\vec{u}_{c|p} - \vec{u}_p)\, |\vec{u}_{c|p} - \vec{u}_p| \tag{5.4}$$

This change is performed in the drag model, creating a new drag model called diFeliceDragFactor. The entire version of the solver is saved under the name BigGaussDPMFoamCdVar. The exact changes to the solver can be found in Appendix B.

5.5 Simulations setup

5.5.1 Simulations and parameters

With the software working and verified, the parameters investigated, the grid generated, and an option added to change the C_d value, the solver can now be used for simulations of the inclined fall pipe. With the simulations, the first three sub-questions will be answered.

As a reminder, they are mentioned here below:

1. Do the simulations predict the same cluster formations and flow effects as in model lab tests?
2. To what extent can the solver predict the rock velocity and production for different angles of the inclined fall pipe and different mixture concentrations in the IFP?
3. What is the influence of various system parameters?

The first two sub-questions will be answered by performing a set of 10 simulations with 2 varying parameters. These parameters are the angle of the inclined fall pipe compared to the horizon and the amount of particles or the particle concentration in the pipe. The values of the IFP's angle will be the same as used in Fontijn's model lab tests: 30, 45, 60, 75, and 90 degrees. The angle is varied by changing the gravity vector, so the domain can stay the same.
The values for the number of particles in the pipe are 275 and 550 particles. The corresponding particle volumetric concentrations over the entire pipe are 1.25 and 2.5%.
These values were chosen so the production of 275 particles would stay in the first half of the domain (0.16 to 0.40 kg/s) for the range of velocities from the model tests (0.45 to 0.8 m/s) and the production of 550 particles would stay in the second half of the domain (0.4 to 0.64 kg/s) for the velocities from the model tests (0.5 to 0.9 m/s).
The simulations were performed with C_{factor} values of 1 and 3, and for the 90 degrees IFP also with a value of 2. The other parameters are given in table 5.6.

Table 5.6: Parameters for the inclined fall pipe simulations.

Parameter	Quantity
Fluid	Water at 20 ° C
ρ_c	998.2 kg/m^3
μ	$1.0035 \cdot 10^{-6}$ m^2/s
Domain	3D
Δx	0.0075 to 0.030 m
Particle	rock
d_p	0.014 m
ρ_p	2650 kg/m^3
Dist. Fac γ	1.0
C_{factor}	1 and 3 (and sometimes 2)
Time	
Δt	$2 \cdot 10^{-3}$ s

The third sub-question will be answered using the same setup as for sub-questions one and two, but other parameters will be varied. The varied parameters are given in table 5.7.

Table 5.7: Varied parameters after the IFP simulations.

Parameter	Original Quantity	Varied Quantity
Δt	$2 \cdot 10^{-3}$ s	$1 \cdot 10^{-3}$ s
Δx	0.0075 to 0.030 m	0.0038 to 0.015 m
Dist. Fac γ	1.0	3.0

The time step (Δt) settings are varied to check whether the time step was not too large during the simulations. By halving the time step and checking for differences, the influence of the time step is checked. There should not be any changes for a time-independent solution. The settings of the time step are set to adjustable by the solver with a standard value of $5 \cdot 10^{-4}s$ and a maximum value of $2 \cdot 10^{-3}s$. As the simulation is mostly running on the maximum value, this one is halved to $1 \cdot 10^{-3}s$ for the dependency test.

The grid size (Δx) is varied to check whether the solutions are grid-independent. The entire grid is remade but with all cells halved in all directions. The cells inside the changed pipe are about 3.8 mm, and outside the pipe, about 15 mm. There should not be any changes compared to the original for a grid-independent solution.

The distance factor (γ) is changed from 1 to 3 to verify whether the reasoning for the distance factor was correct. It will show what happens when the particles are not using a locally averaged velocity for the undisturbed particle velocity but an averaged fluid velocity from a larger area, including the backflow.

5.5.2 Data processing

For each of the sub-questions, different filters are used to end up the required data.
The first and third sub-questions require the rocks to be visualized to compare it to the video material of the model lab tests or to compare it to a case with other settings visually.

Most of the visualizations made in Paraview are made with the same settings and filters. The choice of which data is visualized is done as the last step and can differ per case.
The fluid data is cut halfway through the pipe with the "clip" filter. This way, the fluid data on the clip's location, for example, the velocity, can be visualized. The opacity of the fluid data is always set to 0.5 to be able to see the particles. The coloring can be set to any desired fluid dataset.
The particle data has the representation option set to "Point Gaussian" with the correct radius. The coloring can be set to any desired particle dataset.

The second sub-question requires the rock data to calculate the average velocity and production rate. For a single particle, this is relatively simple, as the velocity can be found by using the filter "PlotDataOverTime," and the velocity can be selected to be plotted over time.

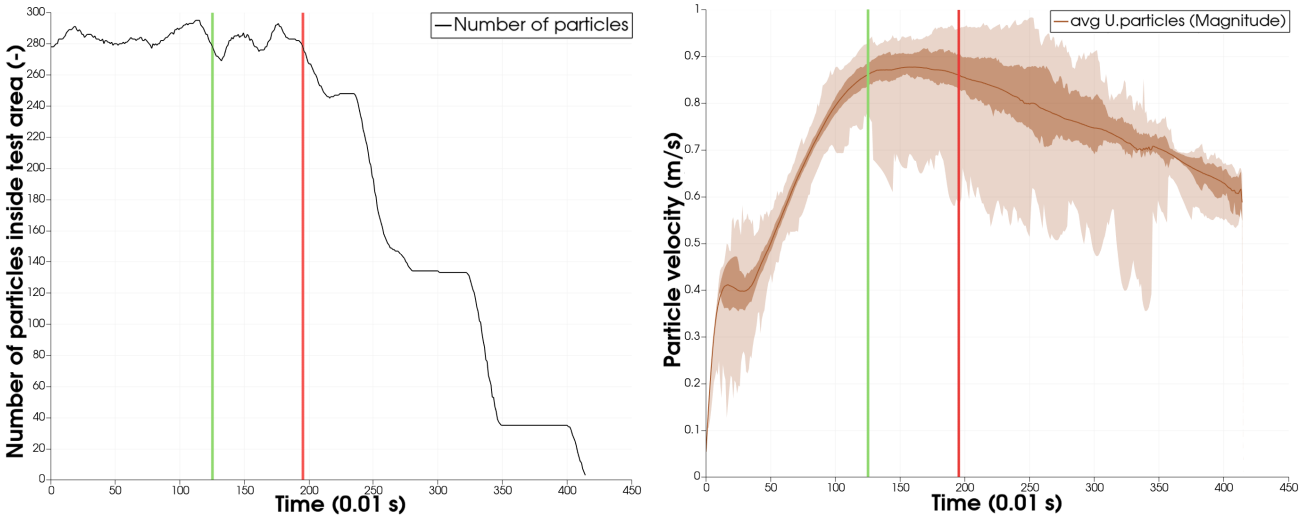
For larger particle groups, filtering out the data before averaging is necessary. The only required data is the data of the particles inside the test area (lower half of the pipe) and only during full production. The

"ExtractCellsByRegion" filter will filter out all particles outside the pipe. The area is set to a block with the starting coordinates set to (0, 0, -1.44), and the size is set to (2, 2, 1.44). This will select only the particles inside the pipe's second half, which is the same pipe length as in the model tests.
Next, the startup and end phases are excluded from these averages. This will be done by finding the time steps at which these phases start and end. The "PlotDataOverTime" filter is used, and the average particle velocity and the number of particles "N" inside the extracted region are plotted in a graph. The number of particles can be seen in figure 5.15a and the particle velocities in figure 5.15b.
With the following steps, as shown for the example data in figure 5.15, the startup and end phase times are determined for each simulation separately:

- The time the end phase starts can easily be found by checking when the "N" value in the figure 5.15a starts to drop. From this moment, no new particles enter the test area from the holding area, the pipe's first half. This moment is marked with a red vertical line around time step 195.

- The time at which the startup phase ends is determined by looking at figure 5.15b and selecting the first time the velocity reaches the same velocity as at the start of the end phase. This is at time step 125 and is marked with a green vertical line.

The data between the green and red lines will be used to calculate the average velocity from data as shown in 5.15b and the production from the data as calculated using the "Calculator" filter. This filter calculates the production with the following equation $Production = N\ avg(|\vec{u}_p|)\ m_p/L$, with m_p as the particle mass (0.0038074 kg for $d_p = 14\ mm$ and $\rho_p = 2650\ kg/m^3$) and L as the pipe length (1.440 m).



(a) The number of particles inside the test area plotted over time.

(b) Mean, q1/q3, and min/max velocities of the particles over time.

Figure 5.15: Examples of the graphs used to determine the startup and end phase times.

5.5.3 Data analysis

The simulation data, both the visual and the average velocity and production values acquired according to the sections above, will be compared to Fontijn's model lab test data to determine whether it is correct. The visual results of the simulations will be compared to pictures of the model tests and to Fontijn's classifications of 'chaoticness' and cluster behavior of the falling rocks.
The velocity and production values are compared to Fontijn's velocity over production data and the velocity over angle data.
In addition to comparing the simulation data to Fontijn's data, the visual results of the clustering process shown by the simulation will be analyzed in detail.

5.6 Limitations

There are several limitations to this methodology due to several reasons. These are categorized into three categories: Limitations due to CFD in general, limitations due to the solver, and limitations due to the experiment data.

5.6.1 CFD

CFD is a method of calculating fluid movements by discretizing a continuous process both over time and over space. The time is divided into time steps. For every time step, a full calculation is made. The larger the time steps, the faster the simulation will be, as there are fewer calculations to be performed. However, large time steps can cause inaccuracies in the simulation as the larger time steps can not represent the continuous nature of the process well enough, and large time steps can cause instabilities in the simulations. Small time steps are required to produce as accurately as possible solutions. However, the smaller the time steps, the longer the calculation times. Since there is only finite time to perform the calculations, the time steps have to be called good enough at some point. The solver automatically selects a time step depending on the velocity of the fluid, and to check whether this is enough, a simulation with even smaller time steps is performed. The changes between those should be minimal and negligible.

The same issue arises with the discretization of the continuous domain into a grid. Smaller cell sizes result in more accurate solutions. However, a smaller grid size results in more calculations. When halving the grid size, the number of cells will be multiplied by 2 in every dimension, multiplying the number of calculations by 8 for the grid alone. The instabilities however are dependent on something called the Courant number (C), which is the fluid velocity times the time step over the size of the cell in 1 dimension ($C = |\vec{u}_c| \cdot \frac{\Delta t}{\Delta x}$). To keep the solution from becoming unstable the time steps will also have to be halved, resulting in a multiplication of the number of calculations by 16 when halving the cell size. As time for the simulations is finite, a grid is created with a grid size that does not introduce large errors but allows to perform the simulations in a reasonable time. A simulation with a smaller grid size is performed to show whether the solutions are still grid size dependent.

5.6.2 Solver limitations

The solver has multiple limitations, some of which follow from simplifications in the physics included, some of which follow from the simplifications in the shape and size distribution of the rocks, and some of which follow from the inability to replicate the experiments exactly as performed.

The solver does include the following particle forces: Gravity, buoyancy (section 3.1.1), viscous drag (section 3.1.2), added mass force (3.1.3), pressure gradient and viscous forces (3.1.4), the wall-particle interaction forces (3.2), the particle-particle interaction forces (3.3.1) and the hindered settling effect (3.3.2). With the compensations of the C_d value, even the higher drag for particles moving along walls and the higher drag force of non-round particles are included. Forces that are not included in the solver are the Saffman lift force (section 3.1.5), the Magnus lift force (3.1.6), the Basset history force (3.1.7), the turbulent dispersion force (3.1.8), the forces due to vortex shedding (3.1.9), and lift forces due to the asymmetric particle shapes (3.4.2). The Saffman lift force and the Basset history force are negligible in the situation of the model inclined fall pipe tests. The other forces, the Magnus lift force, the turbulent dispersion force, the forces due to vortex shedding, and the lift force due to the rock shapes, play a role in the movement of the particles in the rock placement process. The fact that these are not included will influence the correctness of the results. All four of the forces work perpendicular to the slip velocity. The effect that will be visible by missing these four forces will be less sideways movement of the particles over the pipe length.

The particles used in the solver are limited to spherical objects, while the rocks used in the model inclined fall pipe are randomly shaped. By changing the C_d value as discussed in section 5.4, the average drag for a non-spherical particle is considered. All the particles in the simulation experience this average C_d value during the entire simulation. This differs from the drag force experienced by the rocks. Due to the random shape of the particles, the drag due to the C_d value and projected area to the slip velocity experienced by the rocks is different for every rock and changes per rock over time as the orientation of the particle changes. The collisions of non-spherical rocks with walls and other rocks also differ from the spherical particles. The non-spherical shape of the rocks causes the rocks to collide in multiple interaction steps and lose more energy during these collisions.
Another limitation is that the particles in the simulation are all the same size. The interaction model uses

one interaction distance for all particles. By using different particle sizes, the distance for which the particles interact could only be set to one of the sizes, resulting in incorrect collision responses. The difference between one particle size and a scalar of rock sizes is that in the model test with rocks, the particles all have different masses and sizes, causing larger or smaller forces per particle. This change in forces, like the shape, differs for each particle. However, these forces do not change over time.
The result of these two limitations in the solver is that the particles in the simulations are way more homogeneous and experience more homogeneous forces. The rocks from the model tests experience a lot of different forces, meaning they are more prone to moving relative to each other. Larger rocks taking over smaller ones or small rocks slowing down the larger rocks will not be simulated.

The effect of the particles being close to the wall on the C_d value is compensated by changing the C_d value as discussed in section 5.4. This effect should only be compensated for close to the wall. However, the C_d value is independent of the location of the particle. Since, in most cases of IFP simulations, the particles move close to the wall, the correction is used in the simulations, but this is incorrect for (parts of) simulations where the particles do not move along the pipe wall.

Differences between the simulations and the model tests are mostly caused due to the placement of the rocks in the simulations. In the model lab tests, a certain amount of rocks is thrown into the IFP per second, expressed in the production. From this production, a particle concentration in the pipe and an average particle concentration followed.
With this solver, it is not possible to place the particles at the top of the pipe and keep placing new ones throughout the simulation. The ideal case would be to place new particles during the entire duration of the simulation at the top and destroy them after exiting the pipe, only needing limited particles all the time.
An idea was to place all the particles in a holding cell above the beginning of the pipe and let them fall into the test area over time. However, letting the particles start entirely before the test area would mean the first cluster of particles had to create the flow field. As seen in figure 5.12 this uses up a lot of particles. Since this method needs to simulate all existing particles at every time step, it is extra costly to have particles that are used up in this way.
The option used is to start with particles over the entire pipe and let them all contribute to the start of the flow field. This does give the IFP a constant particle concentration from which the production and velocity follow. Because of this change, not the exact productions are used, and the start of the simulation is different from the model tests, but this should not influence the results themselves.

5.6.3 Experimental data

The uncertainties of the experimental data used for the comparisons are discussed in this section.

There is not much known about the used rocks. Only a gradation with a minimum and maximum size is given. It is assumed that these are sieve sizes, but this is not sure. The size distribution between those numbers and the shape factors are also unknown. The size distribution is assumed to be evenly distributed, so the average size of the particles is just taken as the average of the minimum and maximum sizes. The shape factor is not used; however, if known, it would have given the possibility to get a more reliant number for the C_d value from the literature.

Furthermore the velocity results of the model lab tests have big ranges between data points. This could be caused by the randomness of rocks. These ranges might influence the comparison between this data and the simulation data.

6 Results and Analysis

The simulations that are performed, together with the steps taken to filter and compare the data are described in chapter 5. The results are shown in this chapter, divided in three sections. The first section contains the visual and qualitative results of the IFP simulations and a detailed analysis of the cluster formation process of the simulations. The second section contains the quantitative results of the IFP simulations, both the velocity plotted over the production and the velocity plotted over the angle. The third section contains the results of the parameter changes.

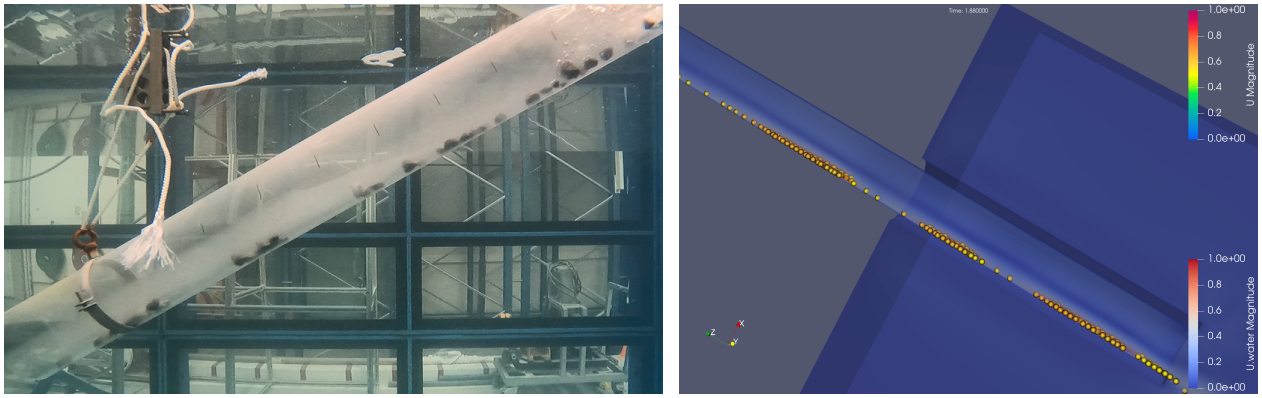
6.1 Qualitative inclined fall pipe simulation results

The process in the inclined fall pipe is simulated with the parameters from 5.6. The visual filters are applied according to section 5.5.2. In this section, first, the visual results are shown next to pictures from the model lab test with the closest production value. Then the qualitative categories are assigned to each simulation and compared to those of the model lab tests.

6.1.1 Visual comparisons

The visual results of the 10 IFP simulations with a C_{factor} of 3, which was found in Appendix D are shown in this section.

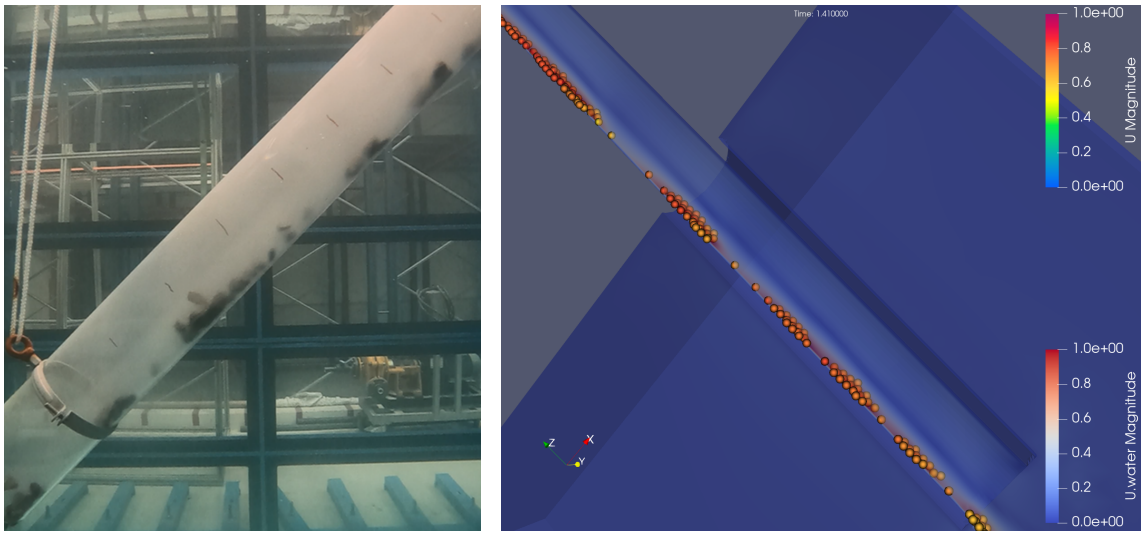
In figure 6.1, the lab test data of an IFP at 30 degrees with a production of 0.16 kg/s is compared to the simulation with 275 particles, a starting particle concentration of 1.25%, and a resulting production of 0.227 kg/s. The processes seem comparable; the particles move along the wall and form clusters. The clusters in the simulation seem to be larger, but that can be accounted to the higher production rate.



(a) Labtest with a production of 0.16 kg/s. (b) Simulation with 275 particles, production of 0.227 kg/s.

Figure 6.1: Comparison between the model lab visual results (a) and the simulation at 30 degrees (b).

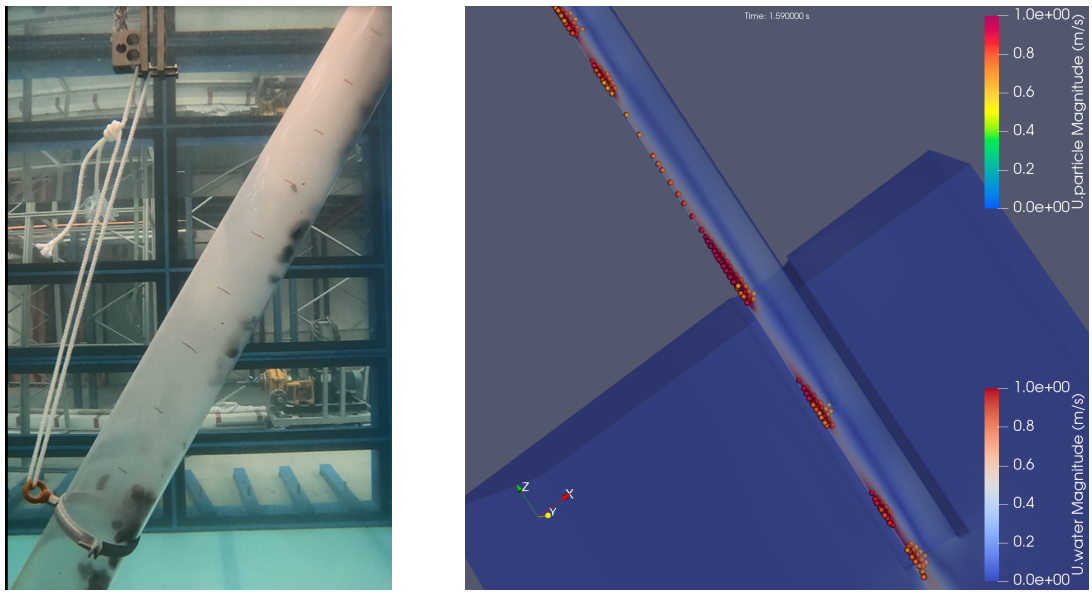
In figure 6.2, the lab test data of an IFP at 45 degrees with a production of 0.32 kg/s is compared to the simulation with 275 particles, a starting particle concentration of 1.25%, and a resulting production of 0.275 kg/s. The processes seem to be very comparable; the particles are moving along the wall and forming clusters. There are multiple clusters of different sizes and some single rocks in between.



(a) Labtest with a production of 0.32 kg/s. (b) Simulation with 275 particles, production of 0.275 kg/s.

Figure 6.2: Comparison between the model lab visual results (a) and the simulation at 45 degrees (b).

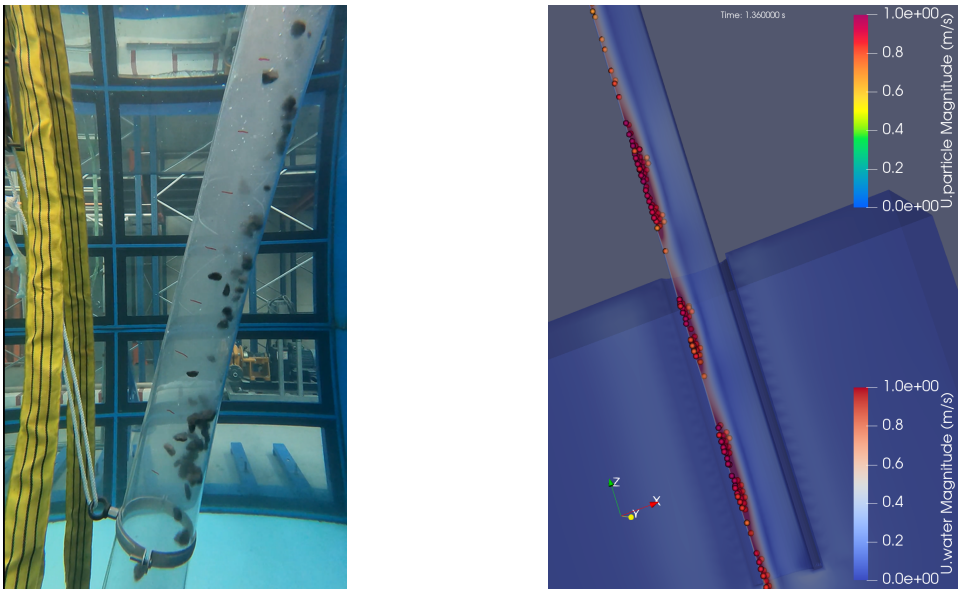
In figure 6.3, the lab test data of an IFP at 60 degrees with a production of 0.32 kg/s is compared to the simulation with 275 particles, a starting particle concentration of 1.25%, and a resulting production of 0.313 kg/s. The processes seem comparable; the particles move along the wall and form clusters. There are multiple clusters of different sizes, and the gaps in between are quite big. Some single rocks floating further away from the wall can be seen in the lab tests; this does not seem to happen in the simulation.



(a) Labtest with a production of 0.32 kg/s. (b) Simulation with 275 particles, production of 0.313 kg/s.

Figure 6.3: Comparison between the model lab visual results (a) and the simulation at 60 degrees (b).

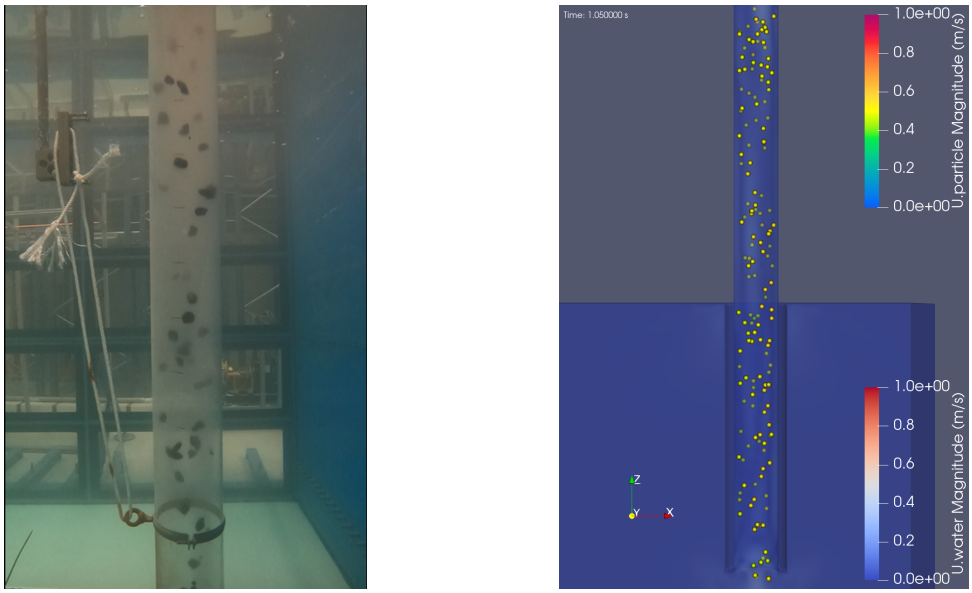
In figure 6.4, the lab test data of an IFP at 75 degrees with a production of 0.32 kg/s is compared to the simulation with 275 particles, a starting particle concentration of 1.25%, and a resulting production of 0.325 kg/s. The processes have visible differences; the particles in the simulation are very compact and stuck to the wall, while in the lab tests, the particles are further spread out over the width of the pipe. When looking at the clusters, the same size of clusters and distances between the clusters can be found.



(a) Labtest with a production of 0.32 kg/s. (b) Simulation with 275 particles, production of 0.325 kg/s.

Figure 6.4: Comparison between the model lab visual results (a) and the simulation at 75 degrees (b).

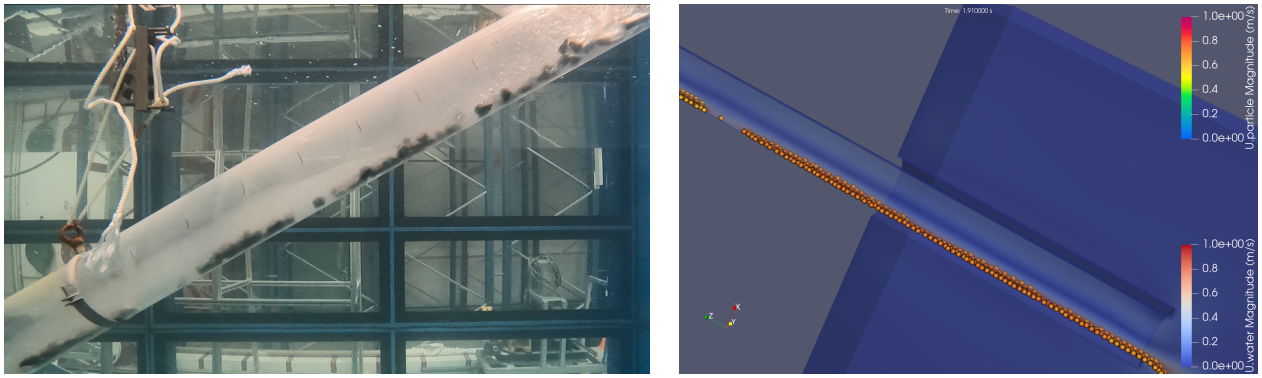
In figure 6.5, the lab test data of a vertical fall pipe with a production of 0.16 kg/s is compared to the simulation with 275 particles, a starting particle concentration of 1.25%, and a resulting production of 0.198 kg/s. These processes look the same but do have differences. The particles and rocks are all in random places in the pipe; however, in the lab tests, the rocks move sideways through the pipe and end up in random locations, while in the simulation, the particles are randomly placed over the domain and only move downwards. Something that does look different is that the simulation seems to have more and smaller rocks than in the lab tests. This might be caused by the rock's gradation and the haze in the tube in the lab tests. The larger particles are better visible, and only a part of the particles is visible through the haze.



(a) Labtest with a production of 0.16 kg/s. (b) Simulation with 275 particles, production of 0.198 kg/s.

Figure 6.5: Comparison between the model lab visual results (a) and the simulation at 90 degrees (b).

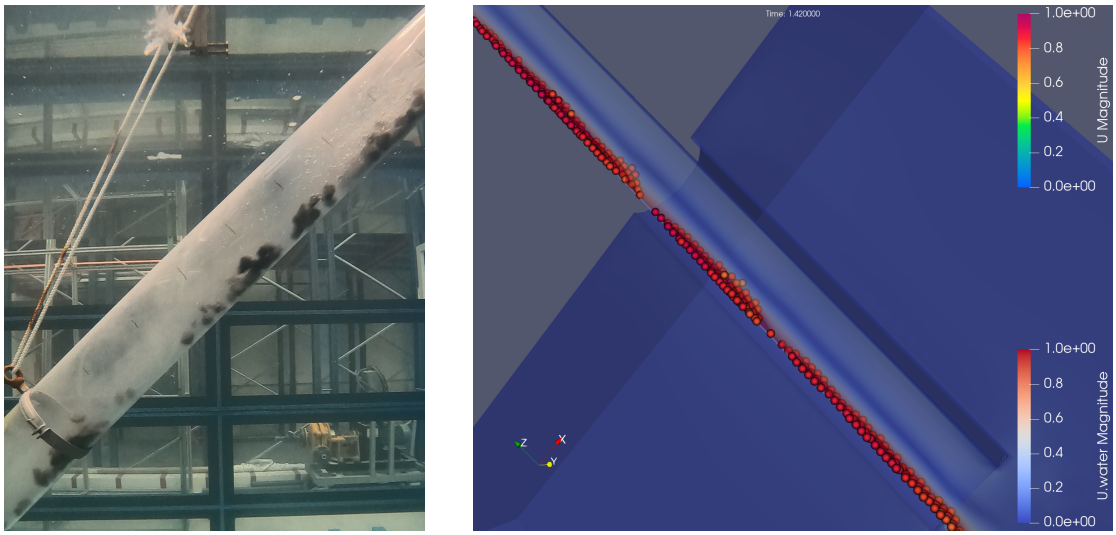
In figure 6.6, the lab test data of an IFP at 30 degrees with a production of 0.48 kg/s is compared to the simulation with 550 particles, a starting particle concentration of 2.50%, and a resulting production of 0.524 kg/s. The processes seem comparable; the particles move along the wall and form clusters. There are fewer gaps between the clusters in the simulation, but there are spots with fewer particles. This difference can be accounted for by the higher production rate in the simulation.



(a) Labtest with a production of 0.48 kg/s. (b) Simulation with 550 particles, production of 0.524 kg/s.

Figure 6.6: Comparison between the model lab visual results (a) and the simulation at 30 degrees (b).

In figure 6.7, the lab test data of an IFP at 45 degrees with a production of 0.64 kg/s is compared to the simulation with 550 particles, a starting particle concentration of 2.50%, and a resulting production of 0.656 kg/s. The processes seem to be different from each other by the sizes of the clusters. The simulation has only a few gaps between a continuous line of particles, while the lab tests show a lot of clusters with very small gaps in between and one larger gap. When considering the small clusters as one cluster by ignoring the really small gaps the size of the clusters does better correspond. The rocks are moving close to the wall in both cases.

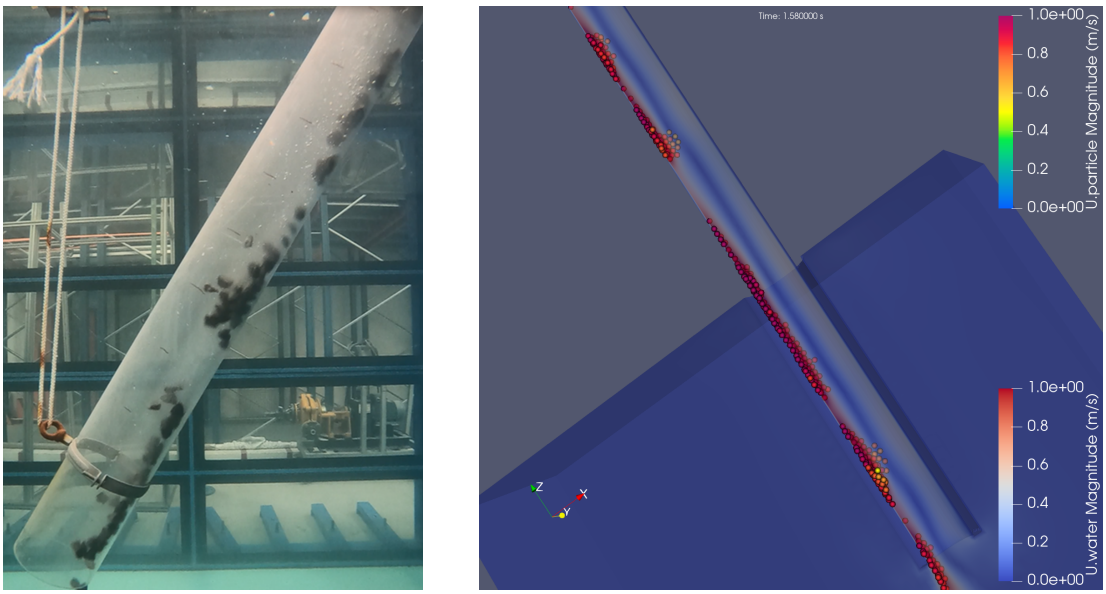


(a) Labtest with a production of 0.64 kg/s. (b) Simulation with 550 particles, production of 0.656 kg/s.

Figure 6.7: Comparison between the model lab visual results (a) and the simulation at 45 degrees (b).

In figure 6.8, the lab test data of an IFP at 60 degrees with a production of 0.64 kg/s is compared to the simulation with 550 particles, a starting particle concentration of 2.50%, and a resulting production of 0.748 kg/s. The processes seem comparable; the particles move along the wall and form clusters. There are multiple clusters of different sizes, with the clusters larger than the gaps in between. In both cases, clusters become large enough that particles at the top of a cluster separate from the cluster as they reach the slower-moving fluid.

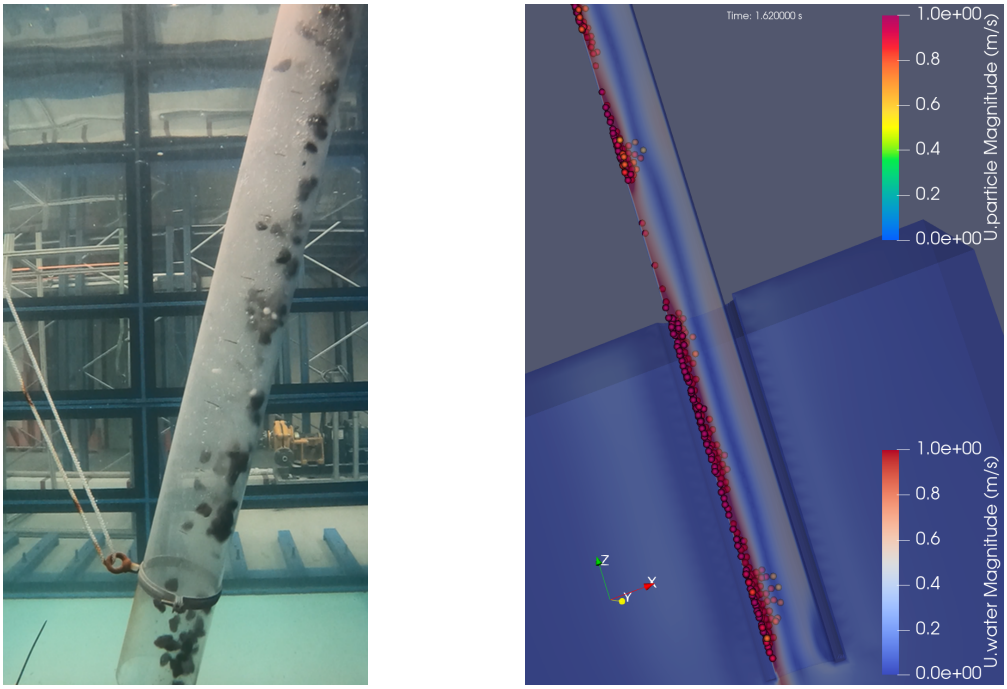
In figure 6.9, the lab test data of an IFP at 75 degrees with a production of 0.64 kg/s is compared to the simulation with 550 particles, a starting particle concentration of 2.50%, and a resulting production of 0.777 kg/s. The processes have visible differences; the particles in the simulation are quite compact and stuck to the wall, apart from some particles at the cluster tops that separate, while in the lab tests, the particles are



(a) Labtest with a production of 0.64 kg/s. (b) Simulation with 550 particles, production of 0.748 kg/s.

Figure 6.8: Comparison between the model lab visual results (a) and the simulation at 60 degrees (b).

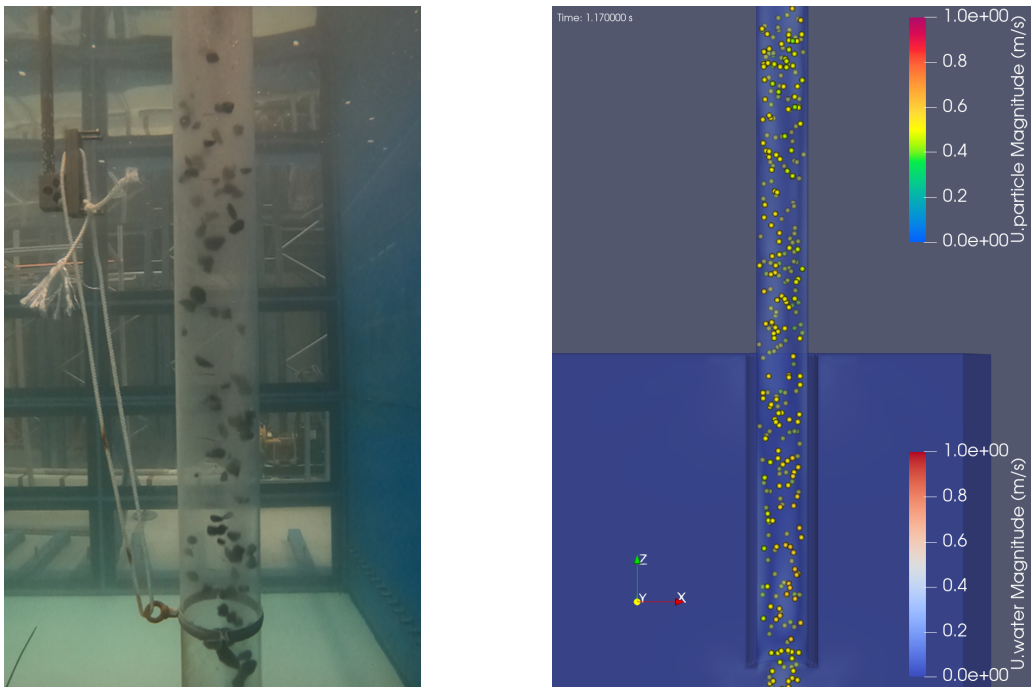
spread out over at least half the width of the pipe. When looking at the clusters, the same size of clusters and distances between the clusters can be found.



(a) Labtest with a production of 0.64 kg/s. (b) Simulation with 550 particles, production of 0.777 kg/s.

Figure 6.9: Comparison between the model lab visual results (a) and the simulation at 75 degrees (b).

In figure 6.10, the lab test data of a vertical fall pipe with a production of 0.32 kg/s is compared to the simulation with 550 particles, a starting particle concentration of 2.50%, and a resulting production of 0.391 kg/s. These processes look the same but do have differences. The particles and rocks are all in random places in the pipe; however, in the lab tests, the rocks move sideways through the pipe and end up in random locations, while in the simulation, the particles are randomly placed over the domain and only move downwards. Something that does look different is that the simulation seems to have more and smaller rocks than in the lab tests. This might be caused by the rock's gradation and the haze in the tube in the lab tests. The larger particles are better visible, and only a part of the particles is visible through the haze.



(a) Labtest with a production of 0.32 kg/s. (b) Simulation with 550 particles, production of 0.391 kg/s.

Figure 6.10: Comparison between the model lab visual results (a) and the simulation at 90 degrees (b).

6.1.2 Classifications of rock behavior

The classifications of the rock behavior from Fontijn and the simulations are shown next to each other in table 6.1. The data of Fontijn is filtered to only the particle gradation of 12-16 mm, and the productions are grouped as one, as the data for the different productions was all the same. The meaning of the letters is as follows:

A Two-layer flow
B Two-layer flow particles in suspension
C Rock clusters
D Some chaos observed
E Total chaos, vertical pipe

After assigning the classifications, the only difference between the simulations and lab tests is between the lab tests and the simulations at 75 degrees. The lab tests show some chaos of the particles at 75 degrees, while the simulations show a two-layer flow with the particles all sticking to the pipe wall.
This difference can be attributed to the missing forces, the Magnus lift force, the turbulent dispersion force, the forces due to vortex shedding, and the shape lift force. All these forces work perpendicular to the flow, which can cause chaotic behavior seen in the lab tests. These forces are less visible at angles lower than 75 degrees since, at those angles, they are (partly) directed against the gravitational force.

Table 6.1: The classifications of the rock behavior per angle of Fontijn for the particle gradation of 12-16mm and the IFP simulations with a particle size of 14mm.

		Angle				
		90	75	60	45	30
Lab tests	Production (kg/s) 0.16-0.64	E	C,D	A,C	A,C	A,C
Simulations	α_p (%) 1.25 2.5	E E	A,C A,C	A,C A,C	A,C A,C	A,C A,C

6.1.3 Clustering process visualization

To enhance the understanding of the clustering process inside the pipe the particles are studied with time steps of about 0.2 seconds during the first 1.65 seconds of the simulation. The particles start randomly distributed at 0.00 seconds and will form clusters within these 1.65 seconds of simulation. The simulation used for these figures is the 60-degree fall pipe simulation with 275 particles, as shown in figure 6.3.
The snapshots are shown 2 times in this section. First, in figures 6.11 to 6.16 where the snapshots are combined and placed next to each other, making the snapshots easier to compare. Then, the figures 6.17 to 6.25 show the snapshots separately from each other and are used as references for the behavior of the particles per time step.
Two subfigures are shown for each snapshot in time. One with the particles colored by their ID number and one by their velocity magnitude. The ID number is given to each particle at the start of the simulation and does not change over time, allowing the particles to be tracked over the different figures. The velocity magnitude shows the differences in particle velocities within particle clusters. The fluid color in both subfigures shows the water velocity magnitude.

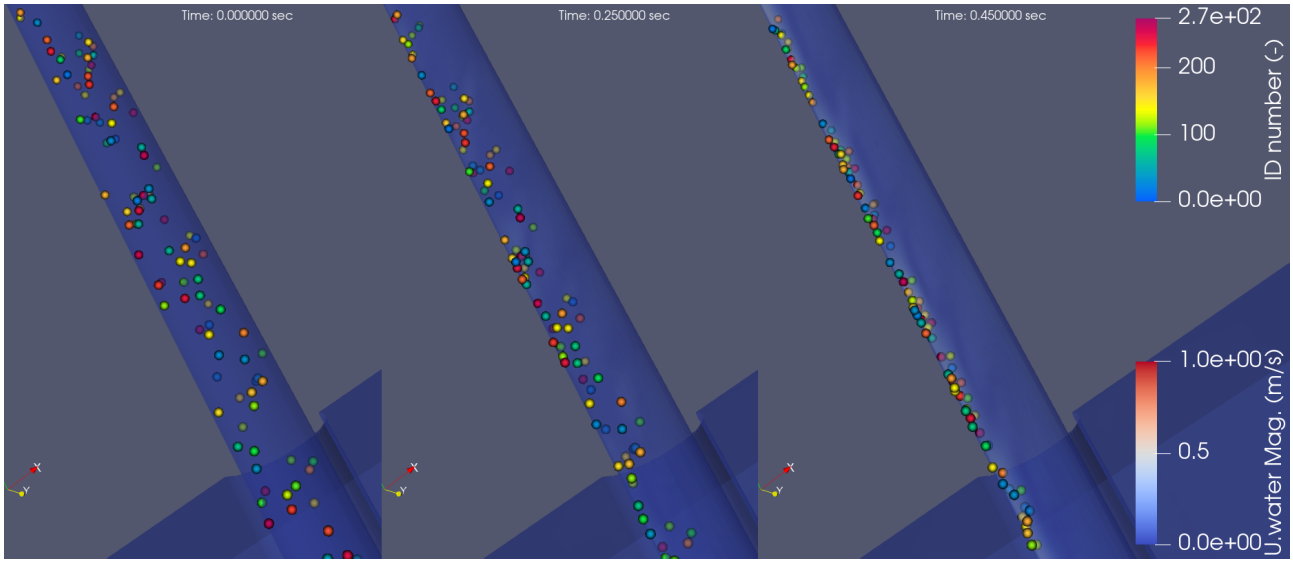


Figure 6.11: The snapshots with the particles colored by ID-number at time 0.00, 0.25, and 0.45 seconds.

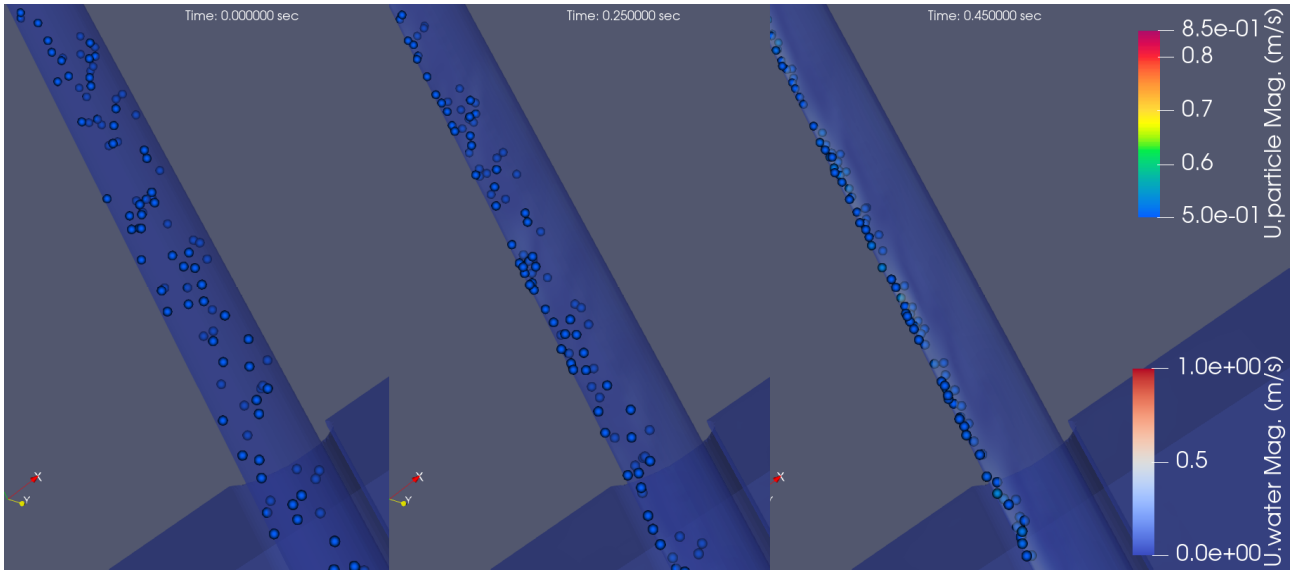


Figure 6.12: The snapshots with the particles colored by velocity magnitude at time 0.00, 0.25, and 0.45 seconds.

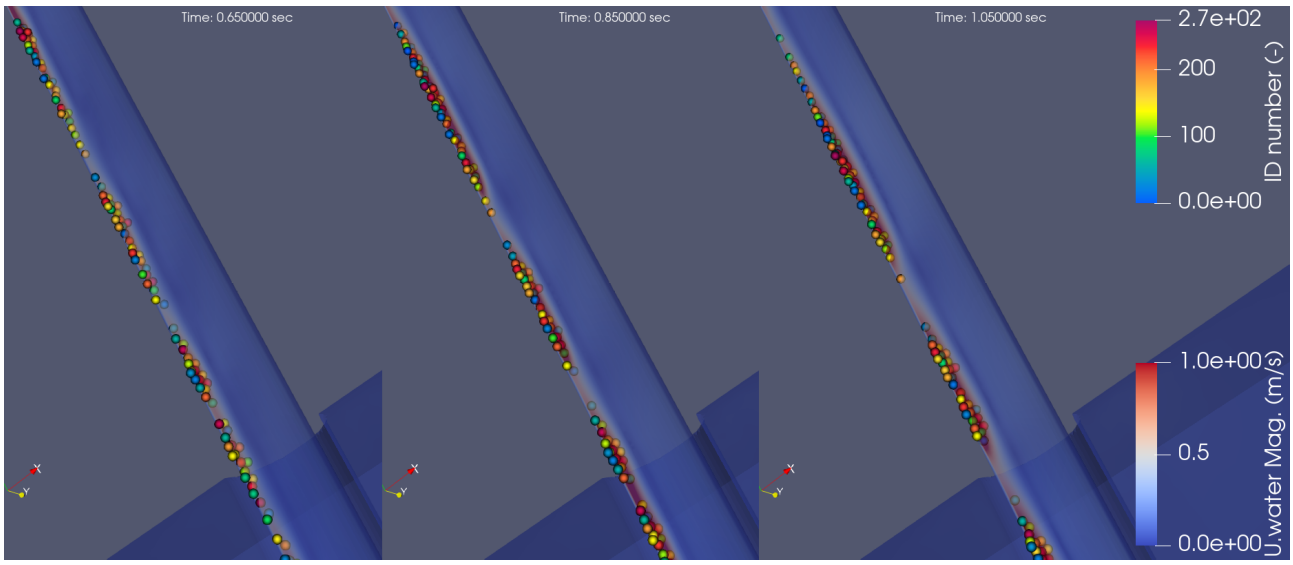


Figure 6.13: The snapshots with the particles colored by ID-number at time 0.65, 0.85, and 1.05 seconds.

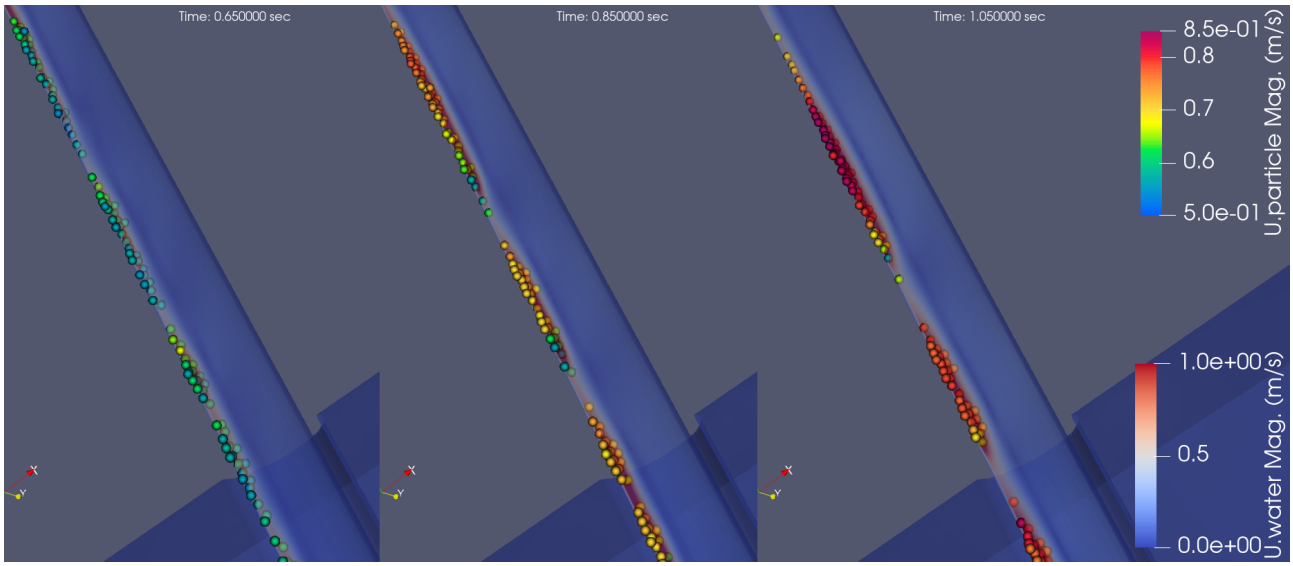


Figure 6.14: The snapshots with the particles colored by velocity magnitude at time 0.65, 0.85, and 1.05 seconds.

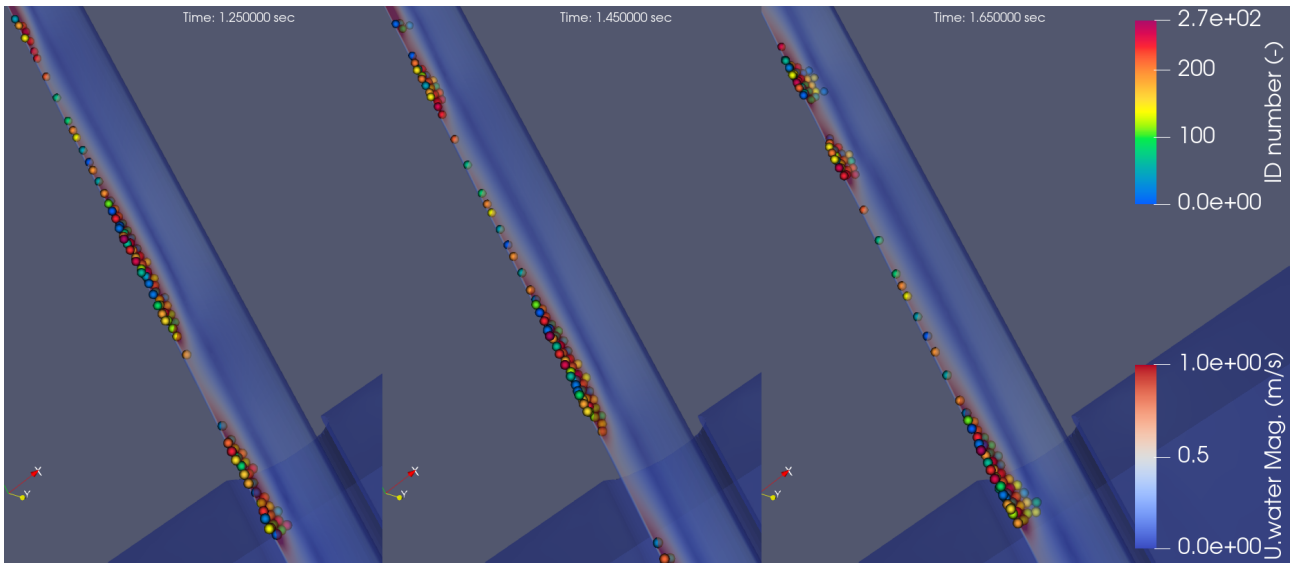


Figure 6.15: The snapshots with the particles colored by ID-number at time 1.25, 1.45, and 1.65 seconds.

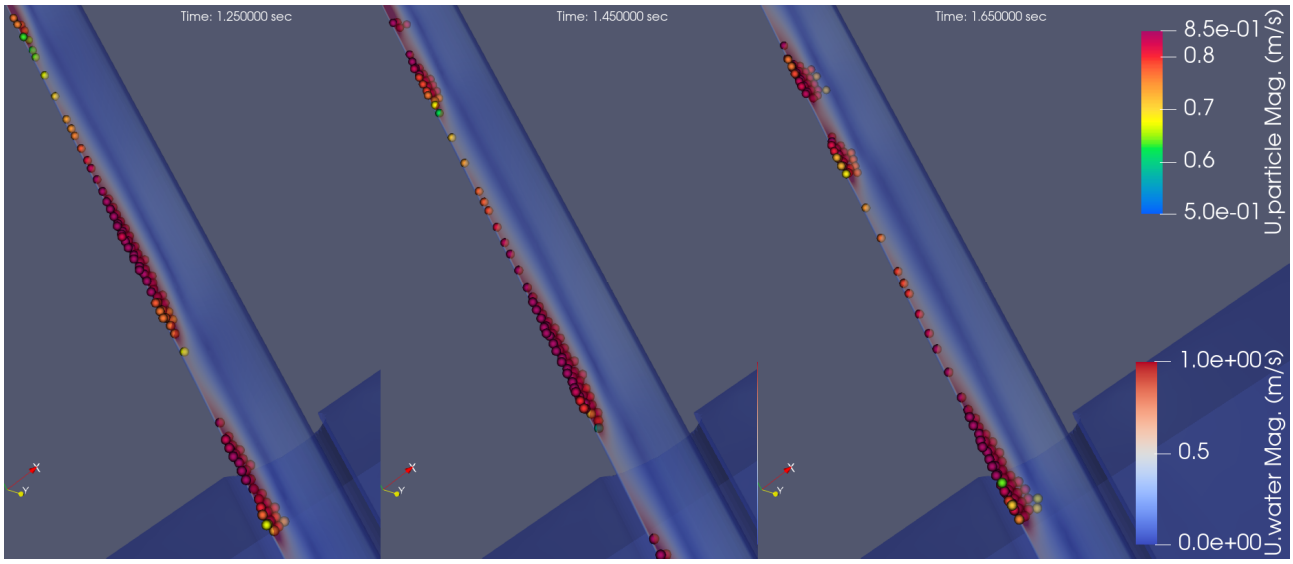
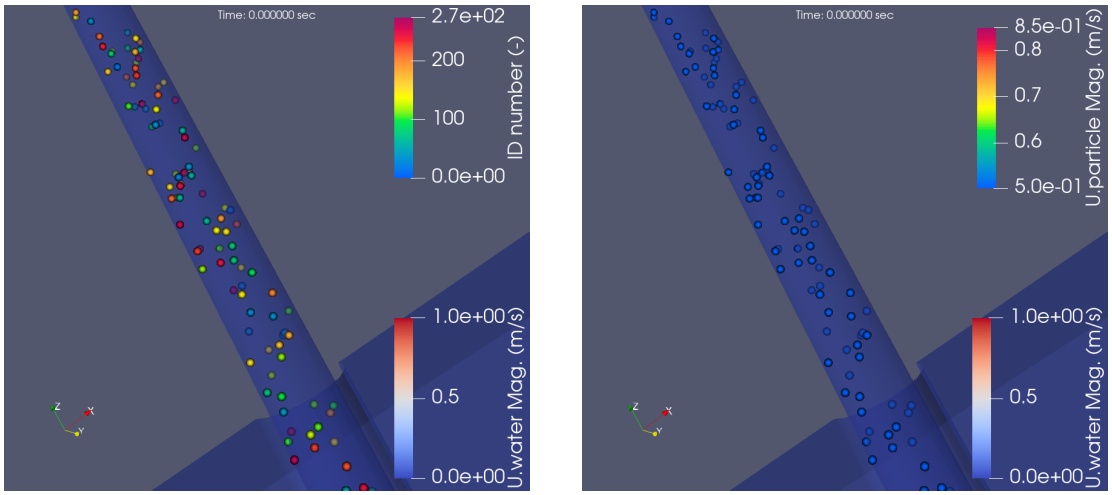


Figure 6.16: The snapshots with the particles colored by velocity magnitude at time 1.25, 1.45, and 1.65 seconds.

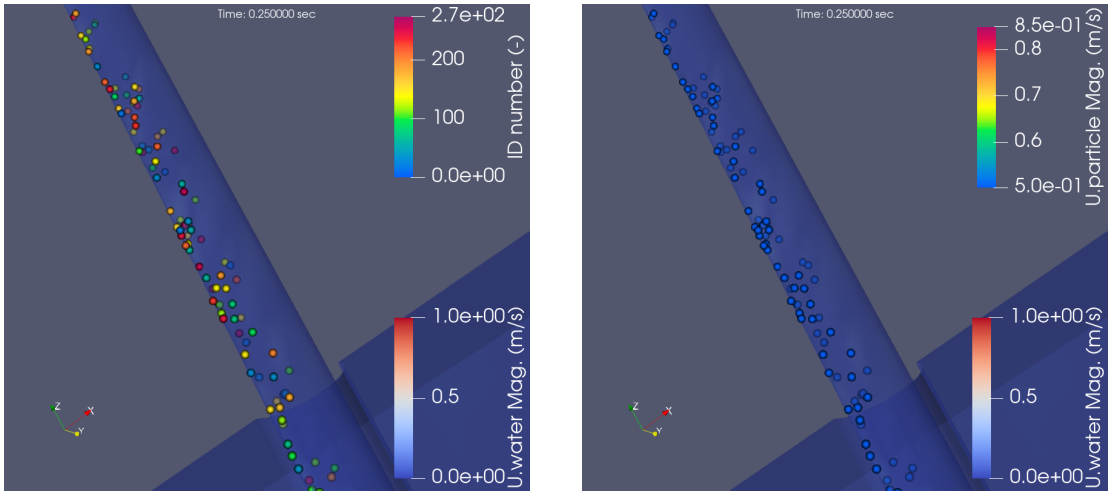
Figure 6.17 shows the particles at the start of the simulation at 0 seconds. The particles are randomly distributed over the IFP, all with a particle velocity of 0 m/s.



(a) (b)

Figure 6.17: The particles at 0 sec, on the left (a) the particles are colored by ID-number, on the right (b) the particles are colored by particle velocity magnitude.

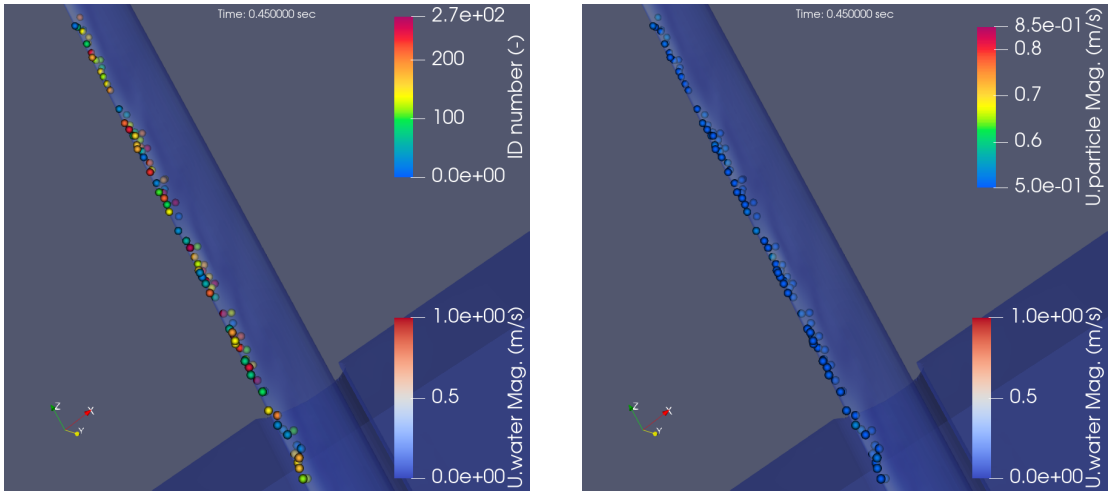
Figure 6.18 shows the particles after 0.25 seconds of simulation. All particles moved down towards the pipe wall. Particles that started closer to the wall did collide with it and are now moving along the wall, while particles that started further from the wall are still falling toward the wall. All particle velocities are under 0.5 m/s.



(a) (b)

Figure 6.18: The particles at 0.25 sec, on the left (a) the particles are colored by ID-number, on the right (b) the particles are colored by particle velocity magnitude.

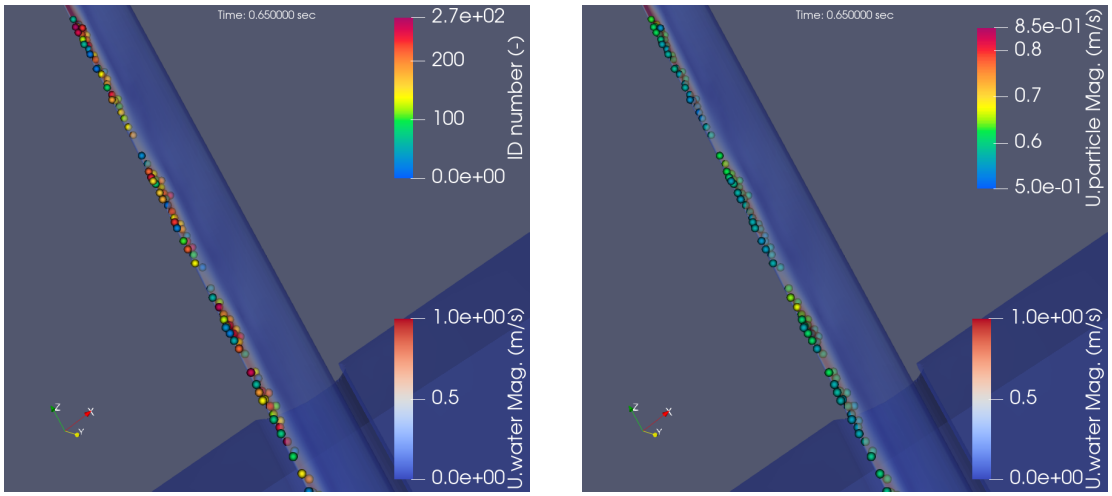
Figure 6.19 shows the particles after 0.45 seconds of simulation. All particles moved toward the wall and formed a layer along the pipe wall. The particle layer is nonuniform, with small gaps between some particles and places where multiple particles are stacked on top of each other. All particles are moving with velocities below 0.5 m/s. The fluid shows the formation of a nonuniform flow downwards along with the particle layer.



(a) (b)

Figure 6.19: The particles at 0.45 sec, on the left (a) the particles are colored by ID-number, on the right (b) the particles are colored by particle velocity magnitude.

Figure 6.20 shows the particles after 0.65 seconds of simulation. The small gaps in the particle layer which were visible at 0.45 seconds have grown, dividing the particle layer into several long clusters. The particle velocities show that the particles at the back (at the up and left side) of the clusters have a higher velocity than the other particles in the cluster. No large difference in particle velocity between the particles in the middle and front of the clusters can be seen. The fluid velocity shows a layer of fluid moving along with the particles, with a couple of higher velocity areas. The fluid near the middle and back of the clusters has a higher velocity than the fluid near the front of the clusters.

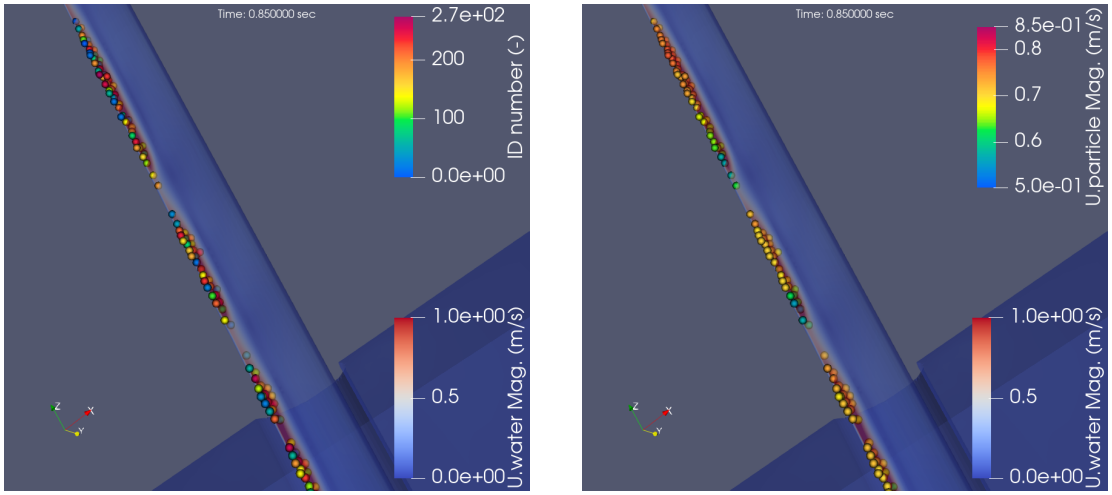


(a) (b)

Figure 6.20: The particles at 0.65 sec, on the left (a) the particles are colored by ID-number, on the right (b) the particles are colored by particle velocity magnitude.

Figure 6.21 shows the particles after 0.85 seconds of simulation. The long clusters from figure 6.20 have contracted to more compact clusters and the gaps between the clusters grew in size. The particle velocities show that the particles in front of the clusters are moving slower, and the particles in the back of the cluster move faster than the particles in the middle of the cluster. The result of these velocity differences is that the clusters will contract even more. The fluid velocity shows clear pockets of faster moving fluid near the particle clusters. When looking between the clusters, the fluid seems to slow down gradually behind a cluster

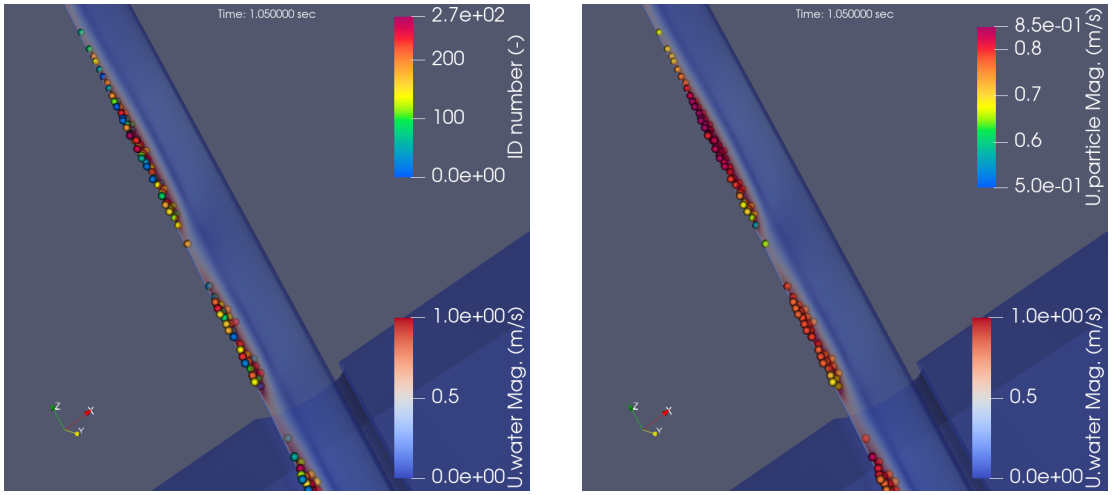
with the slowest fluid in the layer along the pipe wall just in front of the next particle cluster.



(a) (b)

Figure 6.21: The particles at 0.85 sec, on the left (a) the particles are colored by ID-number, on the right (b) the particles are colored by particle velocity magnitude.

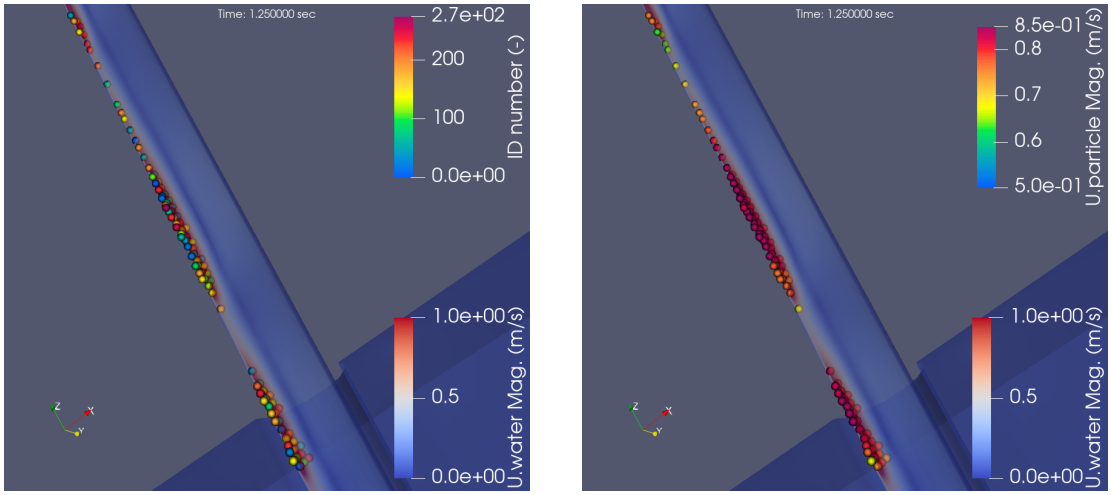
Figure 6.22 shows the particles after 1.05 seconds of simulation. The clusters are a bit more compact compared to the clusters at 0.95 seconds in figure 6.21; especially at the front of the clusters. The particle velocities show that the particles in the front of the cluster are moving slower than the other particles in the same cluster. The particles in the back of the cluster have about the same velocity as the particles in the middle of the cluster, contrary to their velocities at 0.85 seconds, then they were faster than the particles in the middle. For the long cluster at the top of the pipe, the particles following behind the cluster are even moving slower than the particles in the middle of the cluster.
Something interesting can be seen at the front of the large cluster. It has a single particle moving in front of it that moved further away from the front of the cluster compared to 0.85 seconds. The front of the cluster seems to contract away from this particle as the particles at the front are moving slower than the particle, while the particles in the middle of the cluster are moving faster than the single particle.
The fluid velocity shows larger pockets of fast moving fluid near the particle clusters compared to the situation at 0.85 seconds. Between the clusters the fluid velocity looks the same as at 0.85 seconds, gradually slowing down from the back of the cluster towards the next cluster. With the slowest moving fluid of the layer along the pipe wall being just in front of the next cluster.



(a) (b)

Figure 6.22: The particles at 1.05 sec, on the left (a) the particles are colored by ID-number, on the right (b) the particles are colored by particle velocity magnitude.

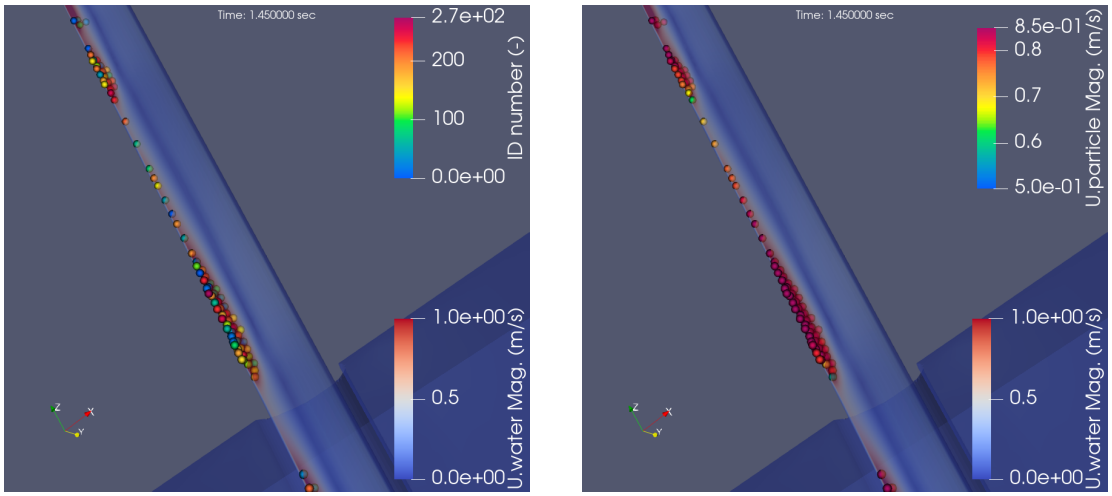
Figure 6.23 shows the particles after 1.25 seconds of simulation. The clusters are about the same size as in figure 6.22, with only the front of the clusters being a bit more compact than at 1.05 seconds. When looking at the particle velocities, the particles at the front of the clusters are slower than the other particles in clusters but are faster than the single particles between the clusters. The gap between the single particle that grew between 0.85 and 1.05 seconds is shrinking again. The fluid velocities show that the fluid has the same fast-moving pockets around clusters at this time as it has at 1.05 seconds.



(a) (b)

Figure 6.23: The particles at 1.25 sec, on the left (a) the particles are colored by ID-number, on the right (b) the particles are colored by particle velocity magnitude.

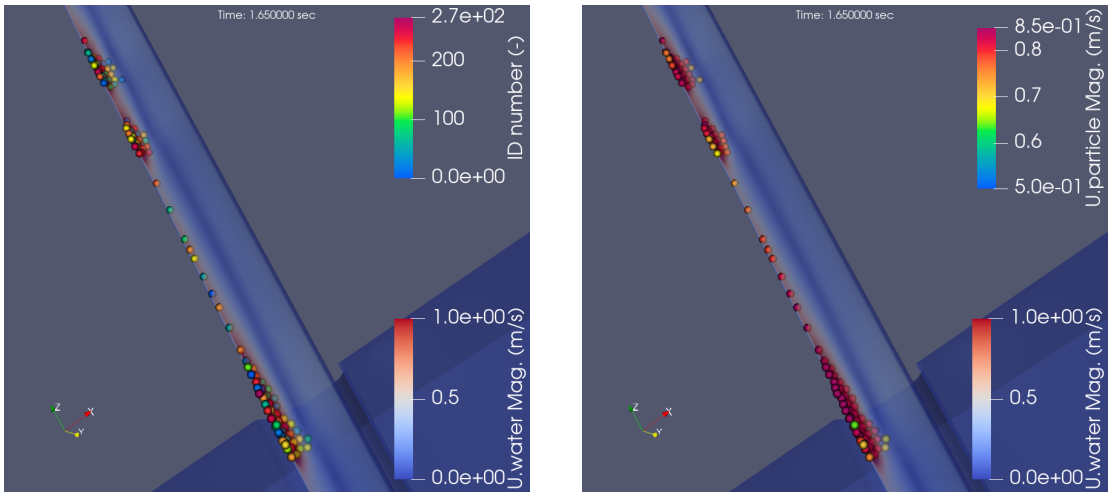
Figure 6.24 shows the particles after 1.45 seconds of simulation. The long cluster caught up to the single particle in front of it and moved away from the single particles behind it. The main body of the cluster only changed at the front; it looks more compact and less pointy than in figure 6.23. Most particles in the clusters are moving with a velocity of 0.85 m/s or higher, and the particles in the front of the cluster have a velocity of about 0.8 m/s. The single particle that was earlier in front of the cluster is moving significantly slower, at about 0.5 m/s. The fluid velocity does not show any changes near the clusters. The two-layer flow is visible in figure 6.24; the flow along with the particles and above it the return flow. A layer with a lower fluid velocity is visible between the two flows.



(a) (b)

Figure 6.24: The particles at 1.45 sec, on the left (a) the particles are colored by ID-number, on the right (b) the particles are colored by particle velocity magnitude.

Figure 6.25 shows the particles after 1.65 seconds of simulation. The front of the clusters show a larger compaction than in figure 6.24, with particles from the entire front being pushed up and to the sides. The particles that are pushed up and to the sides end up in the slower-moving fluid and show a deceleration compared to 1.45 seconds.



(a) (b)

Figure 6.25: The particles at 1.65 sec, on the left (a) the particles are colored by ID-number, on the right (b) the particles are colored by particle velocity magnitude.

6.2 Quantitative inclined fall pipe simulation results

The process in the inclined fall pipe is simulated with the parameters from table 5.6. The particle-selecting filters are applied according to section 5.5.2. First, this section shows the result velocities over production. After which, the result velocities over the angle are shown.

6.2.1 Velocity over production

This section shows the results of the IFP simulations in 5 graphs, one for each angle. The graphs show the resulting velocities over the resulting production rates to compare the results to the data produced by Fontijn and shown in the same manner. The production could be set as an input in the lab tests, and the concentration and velocity followed. In the simulations, the concentration is set as an input from which the velocity follows, and the production is calculated using the concentration and velocity.

In figure 6.26, the average velocities of the IFP simulations at an angle of 30 degrees are shown. For both the original C_d value, the C_{factor} of 1, and the corrected C_d value, C_{factor} of 3, obtained in Appendix D. As can be seen, the velocities are about 0.15 m/s too high for the C_{factor} of 3 and 0.3 m/s too high for the C_{factor} of 1. The C_{factor} of 3 greatly lowers the velocities, but not far enough. The two concentrations' data points show the same steepness as the model lab test results, following the same trend between the velocity and production.

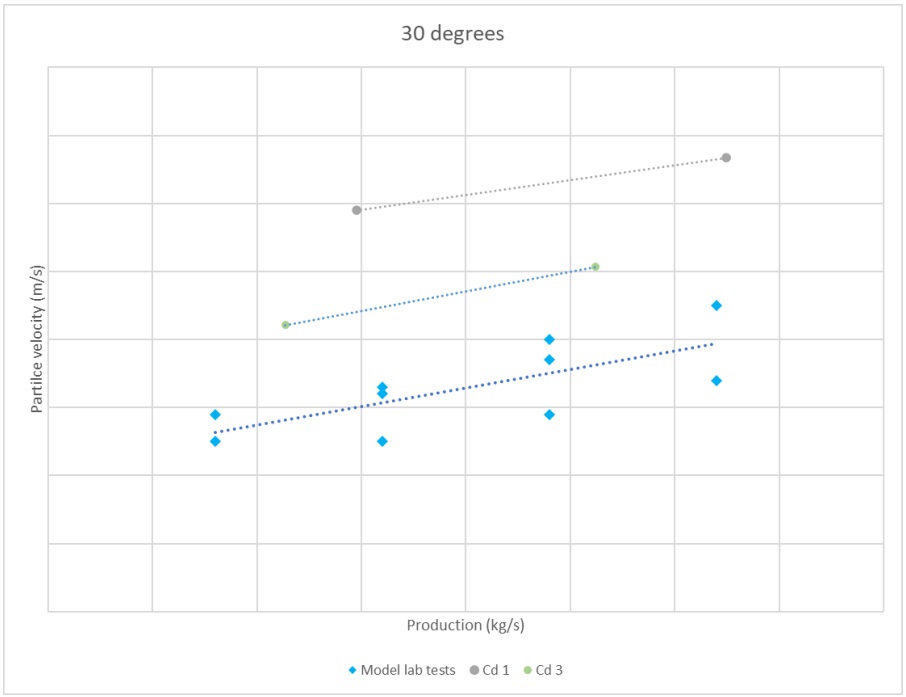


Figure 6.26: The 30-degree IFP simulation data for the concentration of 1.25% and 2.50%, compared to the lab test data from Fontijn.

In figure 6.27, the average velocities of the IFP simulations at an angle of 45 degrees are shown for the C_{factor} of 1 and 3. As can be seen, the velocities are too high for both concentrations. The C_{factor} of 3 lowers the velocities a lot, to just above the data points of the lab tests, but it is still too high. The data points for the two concentrations without the C_d correction show the same trend between the velocity and production as that of the model lab test results. With the C_d correction, the steepness becomes stronger between the data points.

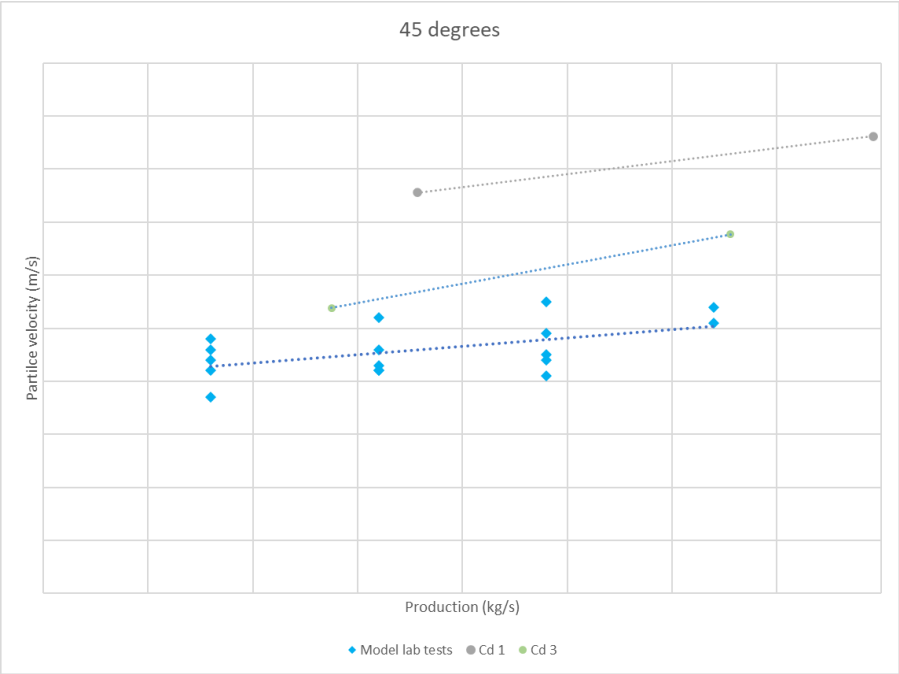


Figure 6.27: The 45-degree IFP simulation data for the concentration of 1.25% and 2.50%, compared to the lab test data from Fontijn.

In figure 6.28, the average velocities of the IFP simulations at an angle of 60 degrees are shown for the C_{factor} of 1 and 3. As can be seen, the velocities are way too high for both concentrations. The C_{factor} of 3 lowers the velocities a lot, but not far enough. The data points for the two concentrations do, however, show the same trend between the velocity and production as that of the model lab test results.

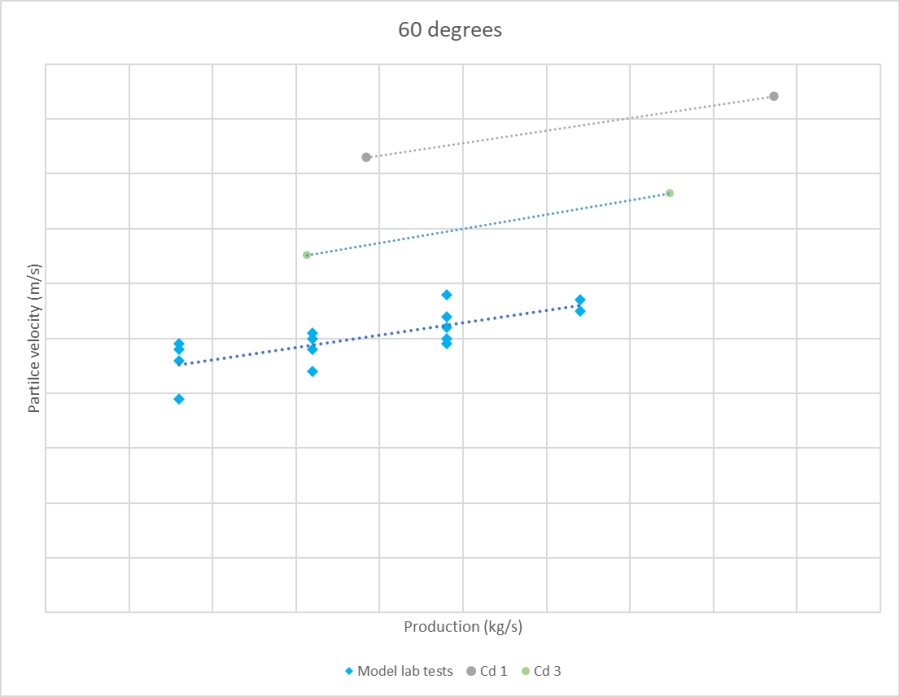


Figure 6.28: The 60-degree IFP simulation data for the concentration of 1.25% and 2.50%, compared to the lab test data from Fontijn.

In figure 6.29, the average velocities of the IFP simulations at an angle of 75 degrees are shown for the C_{factor} of 1 and 3. As can be seen, the velocities are way too high for both concentrations. The C_{factor} of 3 lowers the velocities a lot, but not far enough. The data points for the two concentrations do, contrary to the other angles, not show the same trend between the velocity and production as that of the model lab test results. The velocity and production relationship is stronger than with the model lab tests; the velocity increases faster for

higher production rates.

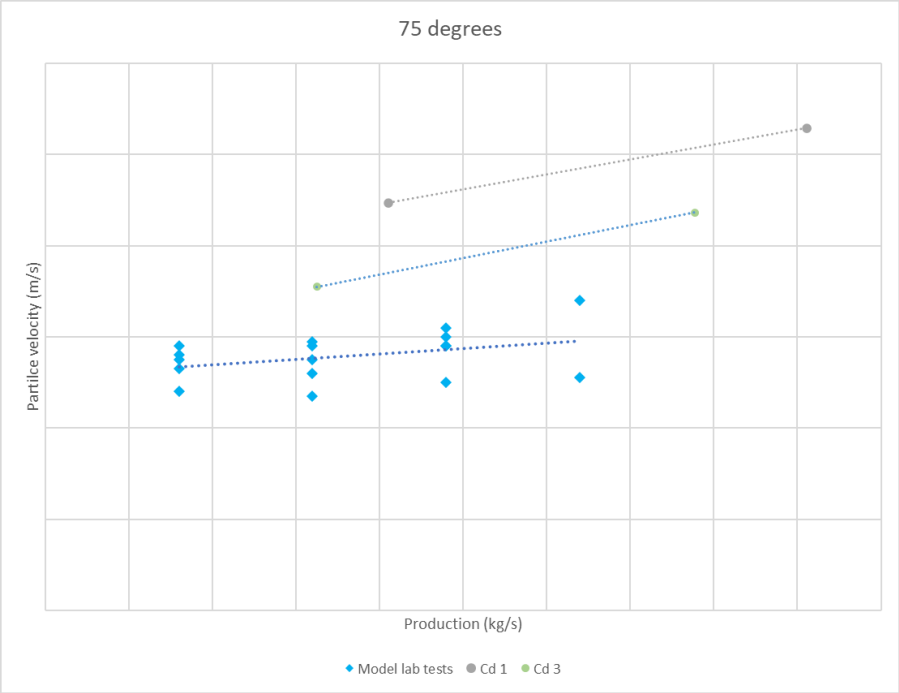


Figure 6.29: The 75-degree IFP simulation data for the concentration of 1.25% and 2.50%, compared to the lab test data from Fontijn.

In figure 6.30, the average velocities of the IFP simulations at an angle of 90 degrees are shown for the C_{factor} of 1, 2, and 3. As can be seen, the velocities of the uncorrected C_d value are way too high for both concentrations. The corrected C_d value, however, brings the velocities under the trend line, and the C_{factor} of 2 lowers the velocities to just above the trend line. Both values for the C_{factor} bring the results within the range of data points from the model tests. An important thing to remember is that the C_{factor} was calibrated to compensate for both the particle moving along the wall and the non-roundness of the particle, and the particles in the 90-degree pipe are not moving along the wall. Thus the C_{factor} of 3 being overcompensated makes perfect sense.

6.2.2 Velocity over angle

Figure 6.31 shows the results of the IFP simulations in a graph with the resulting velocities ordered per angle. The simulations with 275 particles, a particle concentration of 1.25%, and the simulations with 550 particles, a particle concentration of 2.50%, are compared to the model lab test data labeled by production rate. The simulations were performed with the parameters from table 5.6; for the C_{factor}, only a value of 3 has been used.
The velocities of the lower-concentration simulations for the angles up to 75 degrees all lay near the upper end of the lab test velocities and seem to follow the trend of the velocity over angle.
The velocities of the higher-concentration simulations start at 30 degrees, close to the maximum velocity of the lab tests; for the higher angles up to 75 degrees, the velocity shows about the same trend but deviates a bit further from the lab results.
The 90-degree simulations show the same velocity for the lower and higher concentrations, which lies in the lower half of the model test velocities.

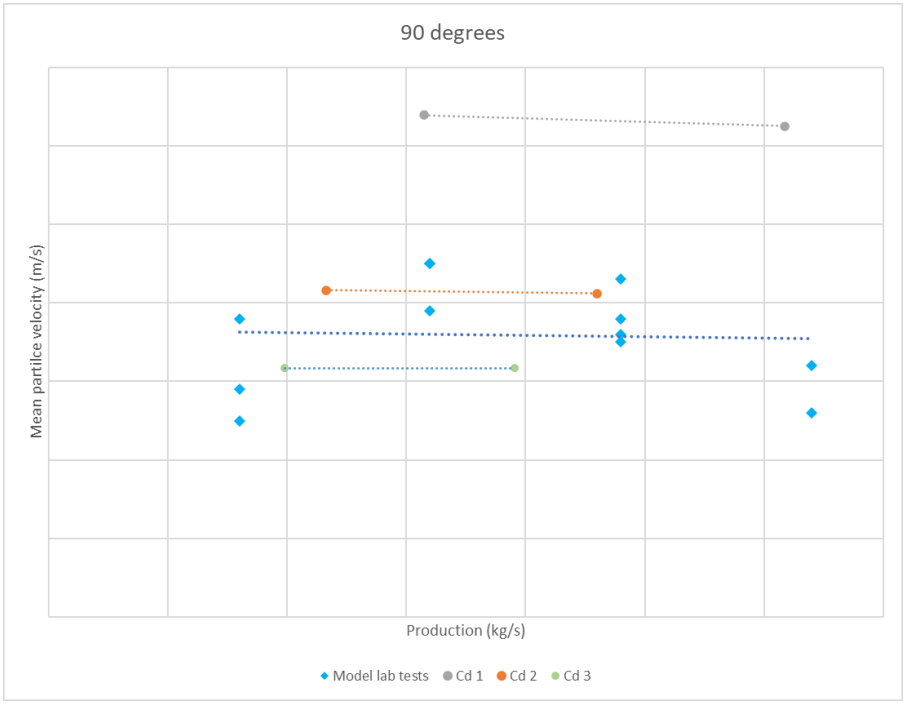


Figure 6.30: The vertical fall pipe simulation data for the concentration of 1.25% and 2.50%, compared to the lab test data from Fontijn.

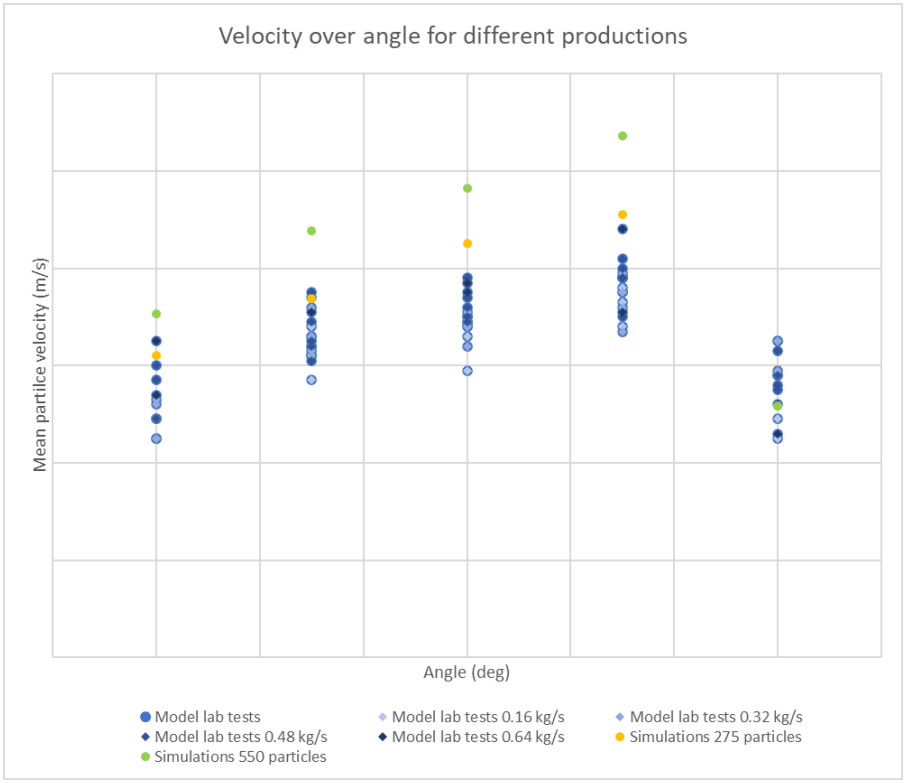


Figure 6.31: The IFP simulation resulting velocities per angle compared to the results of the model lab tests of Fontijn.

6.3 Parameter variations

Multiple comparisons are performed between simulations with the parameters given in table 5.6 and simulations with one of those parameters changed. The system parameters for which the influence is studied are the time step, the grid size, and the distance factor.

6.3.1 Time step

A variation of the simulation with an IFP at 60 degrees and a particle concentration of 2.50% is performed. The parameter varied is the maximum time step. This time step is halved from $2 \cdot 10^{-3}$ to $1 \cdot 10^{-3}$ seconds. Figure 6.32 shows a visual comparison of the original simulation on the left and the simulation with the time step variation on the right. Only a small difference can be spotted between the two. Some clusters are slightly higher in the original simulation, some particles are rolling freely in the time step variation simulation. No large difference can be seen.

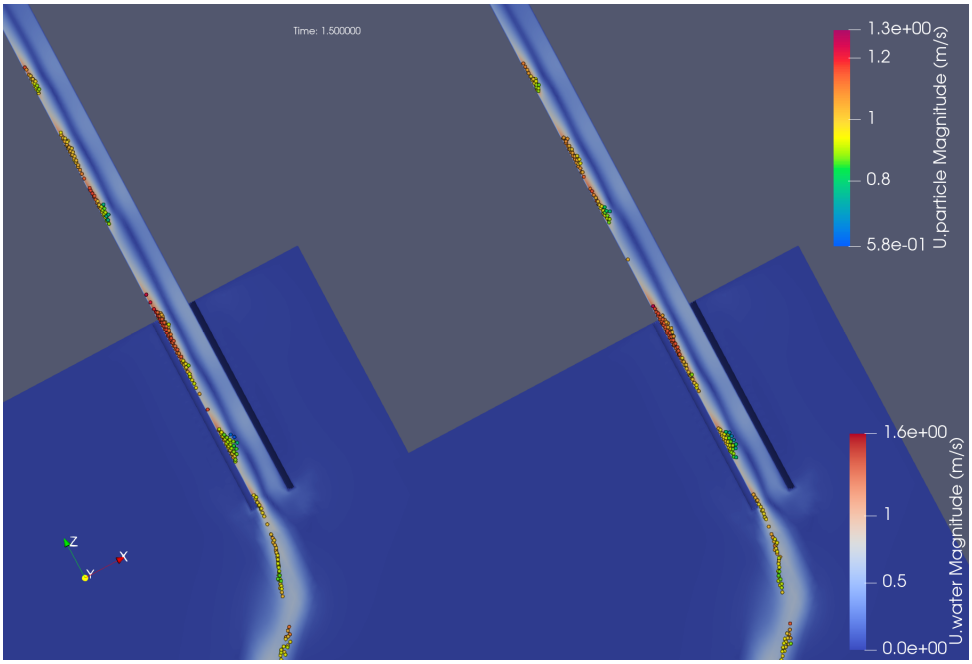


Figure 6.32: Visual comparison for a time step variation. The original simulation on the left uses a time step of 0.002 s, and the simulation on the right uses a time step of 0.001 s.

Figure 6.33 shows the particle velocities of the particles inside the test domain. The velocities start all at the same speed. From 0.2 seconds the minimal and maximal values for the velocity start to differ slightly. After 1 second, the difference becomes a bit larger, but the average velocity only starts to differ from 2 seconds and onwards. The time at which the end-process starts, in which the number of particles inside the IFP drops over time, is around 1.75 seconds. The data gathered from the simulations, which is the average particle velocity up until 1.75 seconds, is not significantly affected by the time step change. The simulation time did grow significantly by the smaller time step, 1 second of simulated time takes about 3.5 hours with the smaller time step compared to 2.2 hours per second for the original simulation.

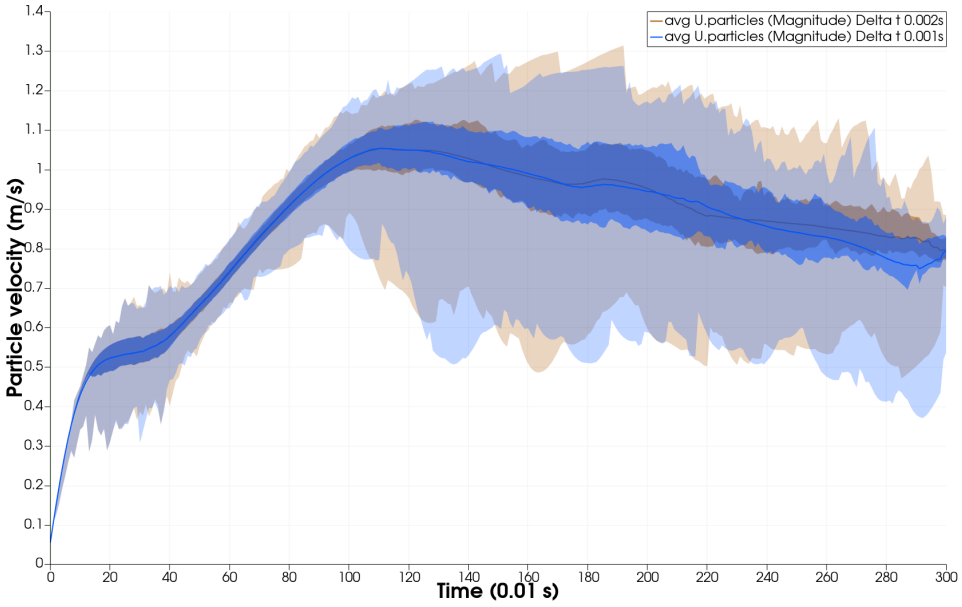


Figure 6.33: Comparison in particle velocities for the original simulation and smaller time step simulation. The line gives the average and the area the min/max and q1/q3 values.

6.3.2 Grid size

A variation of the simulation with an IFP at 60 degrees and a particle concentration of 2.50% is performed. The parameter varied is the grid size. The grid size is halved everywhere. Cells inside the pipe are halved from 7.5mm to 3.8mm and outside of the pipe are halved from 30 to 15 mm. Figure 6.34 shows a visual comparison of the original simulation on the right and the simulation with the time step variation on the left. Only a small difference can be spotted between the two. Some clusters are slightly higher in the original simulation, while the clusters in the simulation with the smaller grid cells are a bit longer. No large difference can be seen.

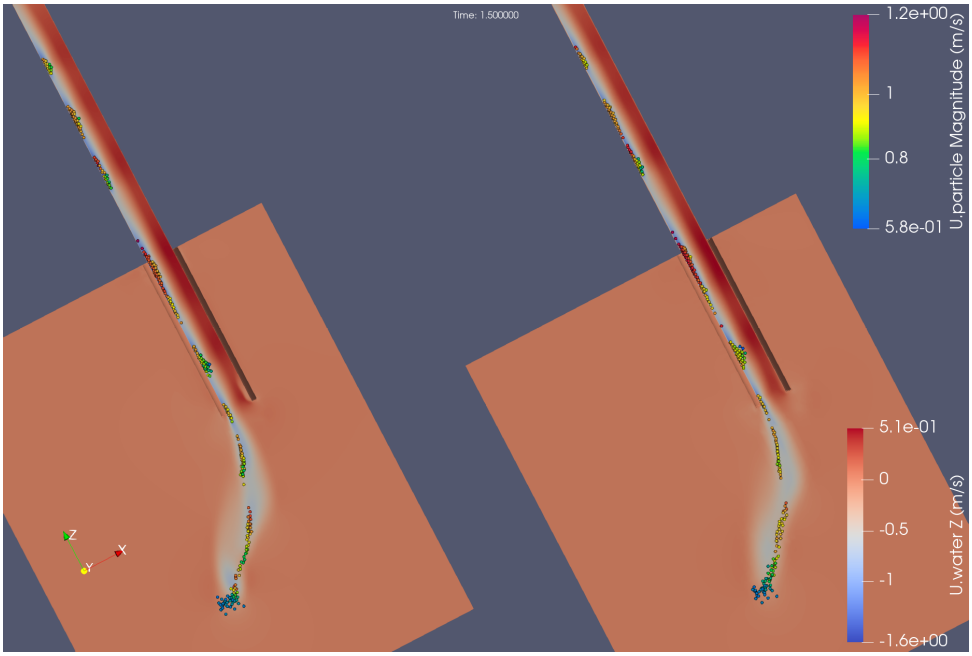


Figure 6.34: Visual comparison for a grid size variation. The simulation with the finer grid on the left uses grid cells of 3.8 to 15mm, and the original simulation on the right uses grid cells of 7.5 to 30 mm.

Figure 6.35 shows the particle velocities of the particles inside the test domain. The velocities start the same. From 1.2 seconds on, average, velocities deviate a small bit. The maximum difference is about 0.03 m/s, 3% of the velocity at that moment, around 1.8 seconds. After the 1.8-second mark, the velocities converge again and oscillate slightly around each other. The average velocity gathered from this simulation is measured between 0.9 and 1.75 seconds. This velocity is only influenced marginally, less than 1%, while the simulation time grew from 2.2 hours to about 80 hours per second of simulated time.

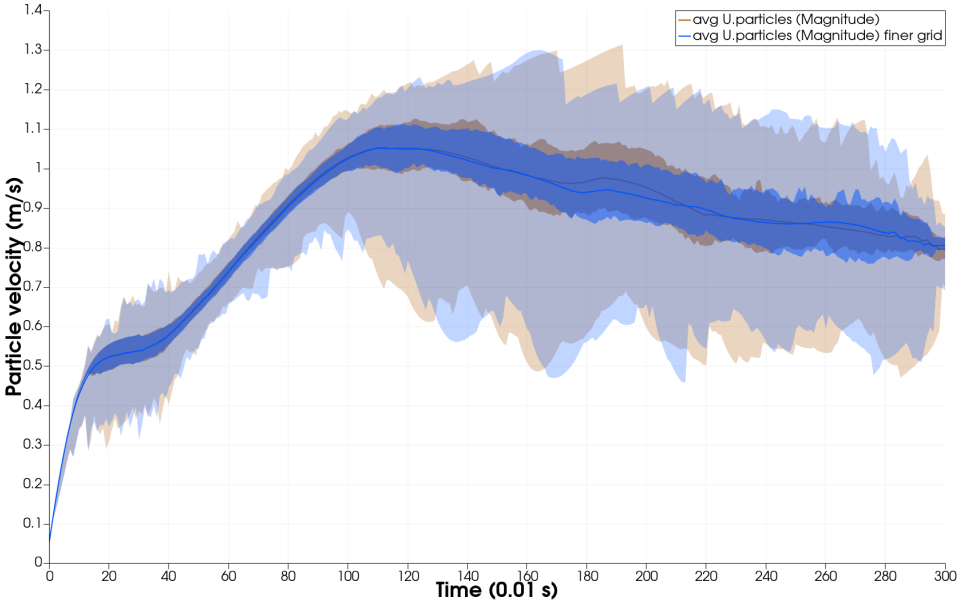


Figure 6.35: Comparison in particle velocities for the original simulation and finer grid simulation. The line gives the average and the area the min/max and q1/q3 values.

6.3.3 Distance factor

The distance factor from the IFP simulations is changed to a value of 3 to see the effects of the higher interaction radius in shear flow. The simulation performed with the varied distance factor is a 75-degree IFP with a particle concentration of 2.5%. Figure 6.36 shows the visual comparison between the simulation with the varied distance factor on the left and the original simulation on the right. The most obvious difference that can be seen is the uniform particle layer along the pipe wall, where the original simulation shows the formation of particle clusters. The kernel function used for the velocity interactions can explain the lack of cluster formations with larger distance factors. The velocities furthest to the edge of the area are contributing the most. When using large distance factors, the local differences near a particle are not used in the drag calculation, and every particle experiences the same interaction forces. With no difference in interaction forces, the particles do not form clusters. A second difference is that the rocks in the original simulation have a much higher velocity.

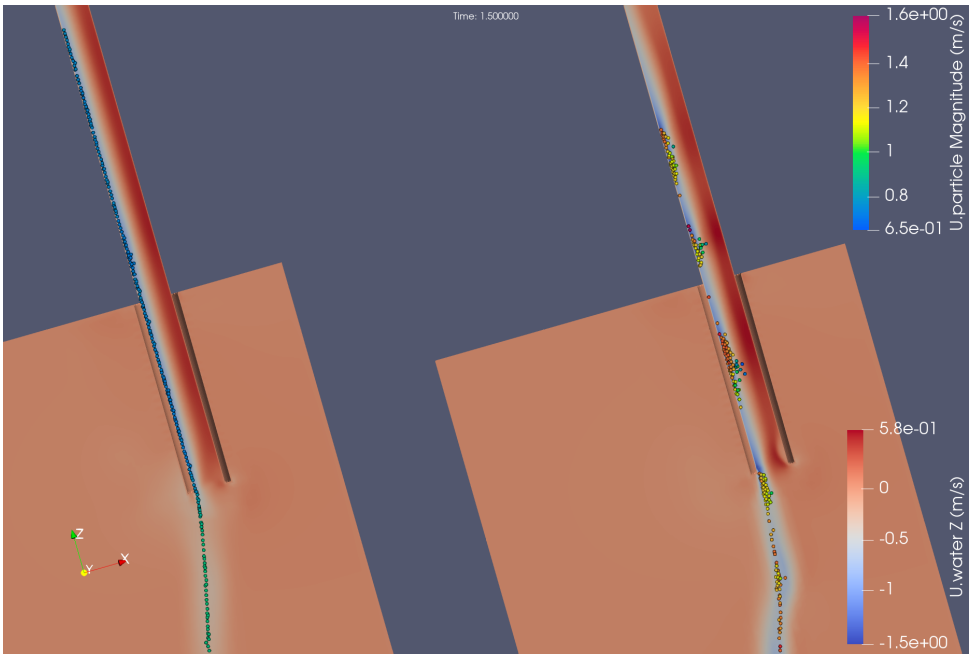


Figure 6.36: Visual comparison for a distance factor variation. The simulation on the left uses a distance factor of 3, and the original simulation on the right uses a distance factor of 1.

The difference in velocity can be seen even better in figure 6.37. The average, minimum-maximum, and q1-q3 particle velocities of both simulations are plotted next to each other. The average particle velocities start similarly, with the randomly distributed particles settling towards the wall. The velocities start deviating from 0.2 seconds, as the particles are gathered along the wall, the fluid starts flowing with the particles, and a return flow starts in the fluid without particles. In the original simulation, the particles accelerate as the fluid within the interaction radius moves along with the particles, reducing the drag. The particles in the simulation with the higher distance factor decelerate a bit, as the interaction radius is larger, and partly use the backflow velocity to calculate the drag force.

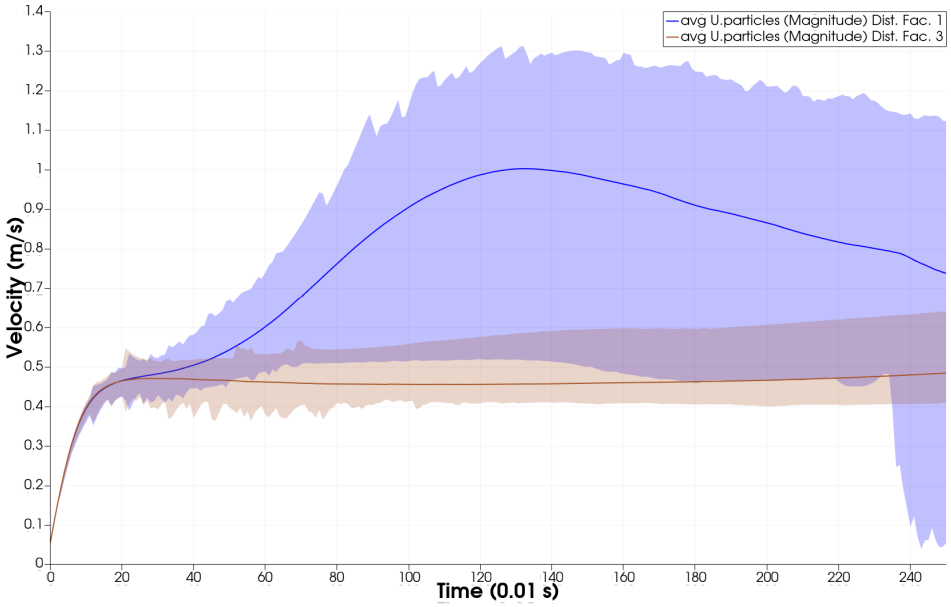


Figure 6.37: Comparison of particle velocities for the original simulation ($\gamma = 1$) and larger distance factor simulation ($\gamma = 3$). The line gives the average and the area the min/max and q1/q3 values.

7 Discussion

The simulation results are shown in chapter 6. In this chapter, the results are discussed. First, the rock behavior inside the IFP will be discussed. First the cluster formations with both the comparison to the model lab results and the observations from the step-by-step analysis of the cluster formation process as simulated. Then the chaoticness of the rocks will be discussed. After the rock behavior, the simulations' prediction capabilities of the velocity and production compared to the results of the model tests are discussed, along with possible reasons for the incorrect particle velocities. Then, in the sections afterward, the implications and significance of the research are discussed, and the research itself will be reflected upon.

7.1 Rock behavior and cluster formations

This section discusses the results regarding sub-question 1: "Do the simulations predict the same cluster formation and flow effects as in model lab tests?" and discusses step by step the cluster formation process according to the simulations. Section 6.1 shows the comparison between the simulation results and the results of the model lab tests. First, a visual comparison is made; after that, the behavior of the rocks is labeled qualitatively and then compared again. Two main observations were made: the first one is that for the angle from 30 to 75, all simulations and lab tests show cluster formations. The second observation is that the simulations of 75 and 90 degrees do not show the chaoticness observed in the lab results. In the following sections, both effects will be described and discussed, whether the physics behind the effects can be explained, and what could be changed to better simulate these effects.

7.1.1 Cluster formations

All simulations at angles of 30 to 75 degrees compared to the lab tests show the formation of rock clusters to some extent, in agreement with Fontijn's results.

To find what causes the formation of these clusters, first, a summary of the formation steps according to the step-by-step visualization of the simulation in section 6.1.3 is given:

- The particles fall from their random starting locations towards the pipe wall.
- The particles reach the pipe wall at different moments in time, causing some to start sliding while others are still falling.
- When all the particles reach the wall, the particle layer on the wall is unevenly distributed. Some places have higher concentrations of particles, and some have lower concentrations. The particle velocities are all below 0.5 m/s.
- From this moment, the clusters start to form. The small gaps in the particle layer grow as the particles at the back of the clusters move faster than the other particles inside the cluster.
- After that, the clusters become even compacter as the particles in the front of the group move slower than the other particles in the cluster, while the particles in the back of the cluster move faster than the other particles in the cluster. The particles in front move slower than single particles between different clusters.
- The particles in the cluster keep accelerating, causing the entire cluster to move faster than single particles. The particles in the back of the cluster are now moving at about the same velocity as those in the middle of the cluster. When particles are too far at the back, they move slower than the cluster, like single particles. The particles at the front of the clusters are still moving slower than the clusters, causing the particles in front to push further back into the cluster. This creates clusters with a large and compact group of particles in the front, with the front being wider and higher than the other parts of the cluster.

- The clusters become too wide at the front, pushing some particles too far away, causing them to slow down in the slower-moving fluid layer.

The summary above describes how the clusters start forming from the difference in concentrations in the particle layer. This finding can be confirmed by the comparison in figures 5.13 and 5.14 in the methodology. Here, the starting condition of an orderly line of particles does not result in cluster formations, while the randomly placed particles do.

The first difference in particle velocities is the back of the higher concentration moving faster than the rest of the particles. These particles have particles in front of them but not behind them. Then all the particles in the cluster accelerate faster than the single particles, except for the ones in the front of the cluster. Those accelerate slower than the single particles but keep accelerating to higher velocities than the single particles.
There is one effect included that is directly dependent on the particle concentration near the particle; this is the hindered settling effect. This effect, however, slows down particles in higher concentrations. Since the particles in higher concentrations are moving faster, this can not be the reason for the formation of clusters.
Another force dependent on local conditions is the drag force. This force depends on the slip velocity, a combination of the local fluid field and the particle velocity. This force can cause the particle velocity differences described above. A clear example of the fluid field can be seen in figure 6.22. Since the particle interaction force pulls along the fluid around it, the fluid gets accelerated along with the particles. The particles still move faster than the fluid, so the most accelerated fluid lags behind the higher particle concentrations, explaining why the particles in the back of the higher concentrations accelerate faster initially. Over time, the fluid within the particles keeps accelerating with the particles, and then the fluid between the particles moves with a smaller speed difference compared to the particles, and the entire cluster, apart from the particles in the front of the cluster, moves faster.
The fluid between two clusters moves the fastest just behind a cluster. The velocity drops gradually when going further away from the back of the cluster, and the lowest velocity is just in front of the next cluster. This lowest point in the fluid velocity, just in front of the next cluster, explains why the particles in front of the cluster initially move slower than the single particles between the clusters. Those experience the highest drag force due to the slowest moving fluid.
The reason why the particles in front of the cluster, after some time, do accelerate to higher velocities than the single particles could be that they are pushed and accelerated by the particles behind them.

That the influenced fluid velocity and the resulting drag forces are the main driving causes for the cluster formations agrees with the simulations in which no cluster formations are visible. These are the simulations with evenly spaced particles, visible in figures 5.13 and 5.14, and with a higher distance factor, in figure 6.36. All particles experience the same reduction in slip velocity in the simulations with ordered lines of particles over the entire tube length. The exception is the first particle, but since that one exits the pipe within the first time steps, this causes no issue.
The larger distance factor in the second case changes the area from which the fluid velocity is calculated. As explained in section 4.3.2, to prevent the fluid velocity from being influenced by the particle itself, the fluid is calculated with a function in which the values of the cells near the edge of the coupling radius have a higher influence than the cells near the particle. When taking the distance factor too large, the fluid velocity will only be calculated with fluid unaffected by the particle cluster. Local differences in velocity do not have any effect on the particles anymore. The number of cells near the edge of the radius is also larger for larger distance factors. This means that even if local velocity differences are present in the cells used to calculate the undisturbed velocity, they contribute a smaller part to the calculated velocity and are averaged out.

The differences between the simulations and the lab tests are mainly the missing forces and the missing size and shape differences between the particles. The four missing forces, the Magnus lift force (section 3.1.6), the turbulent dispersion force (section 3.1.8), the forces due to vortex shedding (section 3.1.9), and the asymmetric shape lift force (section 3.4.2) all move the particles along the pipe width. This does dilute the clusters with more water, especially in the cases with steeper angles, but does not break the clusters apart over the length of the pipe. Section 7.1.2 dives deeper into the effects of these missing forces.
Due to the different gravity and drag forces per particle, particles with different sizes and shapes would create more velocity differences between the particles than the used spheres in the simulations. This means a perfect line of particles will break apart into a random layer of particles and start clustering when the particles are of different sizes or shapes. Thus, different-sized and shaped particles tend to form clusters earlier. The different particle velocities also help particles to break free from clusters.
All three missing effects in the simulations can be seen in the figures of section 6.1.1. A figure showing the lab tests containing more diluted clusters over the width of the pipe due to the missing forces are visible in figure 6.4. A figure showing the lab tests rocks being more prone to start forming clusters is figure 6.6. A figure showing the lab tests containing more broken-free particles is figure 6.10.

7.1.2 Missing chaoticness

The simulations at 75 and 90 degrees do not show the same rock behavior as in the lab tests. The rocks in the lab move in a chaotic way over the cross-section of the pipe, while the particles in the simulation stay in a compact formation along the wall and move straight down the pipe, with the only movement in the directions sideways being caused by the particles forming clusters and pushing each other up.
Chapter 3 describes all forces experienced by a particle in water, not all of them included in the solver. The forces that could move the particles sideways through the pipe and are missing are the Saffman lift force (section 3.1.5), Magnus lift force (section 3.1.6), the turbulent dispersion force (section 3.1.8), the forces due to vortex shedding (section 3.1.9), and the asymmetric shape lift force (section 3.4.2).
The rocks experience particle Reynolds numbers above 1, which is too large for the Saffman force to have any effect.
The Magnus lift force can have an effect, but only if the rock is rotating. Round-shaped rocks can start rotating when rolling along the inside wall. Non-round particles can start rotating while settling due to misaligned gravity and drag forces.
The turbulent dispersion forces for the cases simulated in the IFP can reach in extreme cases a maximum value of about 0.02 N, assuming that the eddies will be of sizes between the rock diameter and the width of the pipe and that the sideways velocity inside these eddies is not larger than 0.5 m/s. This equals an acceleration of almost 6 m/s^2. The maximum time the particles are influenced by the same eddy equals the pass-through time of 0.25 seconds. The direction of the force will be different for every eddy.
The forces due to vortex shedding depend on the velocity of the particles. There is a higher and a lower mode for the shedding frequency. The higher mode can reach up to 300-400 vortices shed over the pipe length for the faster particles. The lower mode is constant over the particle velocity, with about 20 vortices over the pipe length. Each shedding vortex will exert a force on the particle, whereas the vortices with the lower frequencies are generally larger and exert a larger force. Every vortex sheds in a different direction, resulting in forces in different directions
The non-symmetrical shape lift force is the strongest for high slip velocities and differs, both in amplitude and direction, for every particle orientation.

The four forces described above all affect the movement of the particles in a way that could be part of the chaoticness in the lab tests. As these forces are all missing in the solver, it could explain well why the chaoticness is not predicted by the simulations. The side movements are not or less present in the 30, 45 and 60-degree simulations because the sideway forces are directed against the gravity force. The gravity force constantly pulls the particles back to the wall while the four forces are in random directions.

Adding the Magnus lift force, the turbulent dispersion force, the forces due to vortex shedding, and the asymmetric shape lift force to the solver should enhance the prediction capability when simulating the steeper IFP angles. These additions will require changes made to models or new models to be made as described below. The changes are out of the scope of this thesis.

- Adding the Magnus lift force should be possible. All information is available when assuming the particles as spheres. The rotation of the particles is calculated in the solver, and the flow velocity and direction are known.
- The turbulent dispersion force should be possible to add to the solver by using an adjusted version of the StochasticDispersionRAS model already available in OpenFOAM. This model uses the URANS turbulence model already implemented in the solver. The necessary adjustment is to translate the turbulence variables from the grid to the particles correctly.
- No existing model for vortex shedding could be found. Creating a new OpenFOAM model could be based on the turbulent dispersion model, with a force in a random direction tangential to the slip velocity and scaled with the magnitude of the force.
- Adding the asymmetric shape lift force is probably the most difficult as this force is shape and orientation-dependent. With randomly shaped particles tracking of the exact lift force would be very computationally heavy. Approximating the lift force using randomly generated lift coefficients within known limits could be a better alternative.

7.2 Velocity and production

This section discusses the results regarding sub-question 2: ”To what extent can the solver predict the rock velocity and production for different angles of the inclined fall pipe and different mixture concentrations in

the IFP?"
The results in section 6.2 show an overprediction for the velocity, and the from it following production rate, for all simulations at an angle of 30 to 75 degrees. The velocity predicted with the simulations of the 90-degree fall pipe shows a better agreement with the lab results. The 90-degree fall pipe simulations and the 30 to 75-degree fall pipe simulations will be discussed further in two separate sections as the process is quite different. First, the 90-degree or vertical fall pipe simulations will be discussed, then the inclined fall pipe simulations.

7.2.1 90-degree fall pipe simulations

Figure 6.30 shows that the simulation results with a C_{factor} of 3 fall within the uncertainty of the lab test results. However, the used C_{factor} of 3 is too high as it contains the 1.6 factor for the effect of the particles moving along the pipe wall, which the particles do not do in the 90-degree fall pipe. When compensating for this and using a C_{factor} of 2 (close to 1.9), these results also fall within the uncertainty of the lab results. The reason the results for both C_{factor} values fall within the uncertainty of the lab results is not because the difference in velocity between the values changes so little; on the contrary, the velocity changes by about 20%. The range of velocity values as results of the lab tests is large, from 0.45 m/s to 0.65 m/s. The average velocity of the lab results is about 0.56 m/s.

The almost constant velocity value that the simulations and the lab result trend line show can be explained by the process being nothing more than a (hindered) settling case. With the particles of the same size, the settling velocities will stay the same, and the hindered settling effect will not affect the velocities significantly as the particle concentrations are in the single digits. The simulation results being almost constant values is thus not a big surprise; the lab results showing this exact trend with only 12 values and such a big uncertainty is a larger surprise. The large differences within the lab results' velocity values can be explained by the variety of used rocks' shapes and sizes.

The value for the C_{factor} needed for a simulation to mimic the average velocity of 0.56 m/s would be about 2.5 or 2.6, much larger than the value of 1.9 gained from Appendix D when not adding the wall effect. This higher value would better fit the values found in literature according to section 3.4.1.

7.2.2 30 to 75-degree fall pipe simulations

The simulations with a 30 to 75-degree fall pipe all overpredict the velocity at the same production level. The correction performed using a C_{factor} of 3 instead of 1 results in lower velocities, but a rather large difference in velocity with the lab test results remains. The difference in velocity shows that the correction with the C_{factor} calibrated as in Appendix D is insufficient to predict the velocity of the particles correctly. The correction with the C_{factor} is an attempt to correct for the shape of the particles and the effect of the particles moving along the wall. Three reasons for this difference in velocity could be thought of:
The first reason could be that the missing forces mentioned in section 7.1.2, by influencing the width of the clusters, create a sparser particle density and, in that way, cause a larger backflow between the particles. The same number of particles would, when influencing a larger volume of fluid, result in less interaction force being transferred from the particles to each liter of fluid, and thus the fluid will be less pulled along than in compacter clusters. This means the deviation would depend on the width of the clusters, which is dependent on the angle of the pipe. The steeper, the wider, and sparser clusters.
The second reason could be that an incorrect C_{factor} was used; uncertainties in the lab results used for the calibration could have resulted in an incorrect value. Section 7.2.1 also showed a mismatch in the C_{factor}. All resulting velocities would be slower with a larger C_{factor}. The drag force scales with the C_{factor} and the fluid velocity with the second power.
The third reason could be the use of an incorrect distance factor. Section 6.3.3 shows the results of a distance factor variation. As can be seen, a value that is too large for the distance factor could cause too low velocities in the simulation. However, the distance factor may have been slightly higher, dropping the velocity by the 25% difference. This can not be known until a more in-depth study of the correct distance factor for shear flows is performed.

To compare the simulation and lab results of figures 6.26 to 6.29, the slopes and offsets of the trendlines are shown in table 7.1. It can be seen that the slopes are about the same for some of the angles but not for all of them. The largest differences can be seen at 45 and 75-degree angles. For the angle of 60 degrees, the slopes differ the least. The inconsistency means the velocities of the simulations can not be used to predict the velocities in the lab. The correlations between the velocity and both the production and angle are the same

for the simulations and the lab results, meaning the slope between two points is either positive or negative for both the simulations and lab results, for the results found. This correlation allows for qualitative comparisons between the simulations and lab results.

Table 7.1: A overview of the slopes and offsets of the trendlines through the lab and simulation results of figure 6.26 to 6.29.

Angle (°)	C_{factor} 1, 3 or lab result	Slope (m/kg)	Offset (m/s)
30	C_{factor} 1		
30	C_{factor} 3		
30	lab		
45	C_{factor} 1		
45	C_{factor} 3		
45	lab		
60	C_{factor} 1		
60	C_{factor} 3		
60	lab		
75	C_{factor} 1		
75	C_{factor} 3		
75	lab		

7.3 Implications and significance of the research

In this thesis, Nieuwboer's OpenFOAM solver is used with minimal adjustments to simulate the process inside an inclined fall pipe. The solver was developed to simulate the spilling process in the rock cutter, which has quite different features than the IFP rock placement process. The main driving force for the rocks was the pressure difference and rotational forces, while in the IFP the main driving force is the gravitational force. Furthermore, the IFP has higher shear rates close to the rocks than in the rock cutter.
The visual results show that the cluster sizes are predicted quite well, especially for angles from 30 to 60 degrees. One of the reasons for using an IFP over a vertical fall pipe is to gain clearance between the vessel and the location where the rocks have to be placed, so angles of 60 degrees and up are often used. The predicted rock clusters could be used as input for a dispersion model calculating the spread and offset as performed by Hogenhuis [31]. The input would enhance the prediction capabilities of that model, as it used a constant stream instead of a flow with rock clusters. Instead, the solver would also be able to perform the entire simulation when a dispersion model is added to the solver itself, predicting how far and where the rocks will end up on the sea floor for different variables of the IFP.
A valuable insight from studying the rock cluster formation process is the deduction that the main cause for the particle cluster formation is the local velocity differences. The variety in rock sizes and shapes may contribute to the cluster formations, but the particle clusters still form when uniform-sized particles are used, as seen in the simulations. Changes to the IFP design to have the water flow along with the rocks could lower the local velocity differences and, in turn, the cluster formations.
The results of this thesis also show that the solver does overpredict the velocity but with the results following the velocity over production and velocity over angle trends. This means that the solver can be used to predict the changes in velocities when different angles or production rates need to be compared.
The above applications and insight show how the knowledge of the prediction capabilities of the solver could be utilized as a stepping stone to enhance the knowledge of the IFP. In turn, the rock placement process can become more efficient, which saves resources and time in operations.

For this thesis, much time was spent using Nieuwboer's solver to gain insights into the IFP rock placement process. This resulted also in insights about the solver, the most important of which is that the solutions depend heavily on the distance factor γ in shear flows. It is important to know that this system parameter has to be tuned correctly.

7.4 Reflection

The limitations of CFD in general, the used solver, and the used experimental data are discussed in section 5.6. Furthermore, the factors that could impact the results are discussed in every section of this chapter. In

this section, the largest uncertainties and the strengths and weaknesses of the methodology are discussed.
The largest uncertainty is caused by the lack of knowledge about the rocks used in the lab tests. The known size of the rocks is the minimum and maximum size given by a supplier, and most often is the sieve size, but even this is not certain. The distribution between those values is also not known. The impact is that the IFP simulations could be performed with incorrect-sized particles influencing the results. When looking at the visual comparisons, the rocks seem to be larger than the particles in the simulations. According to the lab test results, larger particles will result in lower particle velocities in IFP tests with multiple rocks.
To mitigate these issues in the future, lab tests could be performed with particles like marbles. Then the simulations and lab tests could be compared directly. These tests no longer compare to real-life IFP usage but could be used to validate and improve the solver further. Another option is to count and weigh the batch of particles before the lab tests. Then the average weight and if the density is known, the volume is known, which makes it easier to use particles of a representable size.

A weakness of the methodology is that the distance factor was only validated for homogeneous flows. The large influence of the shear flow was only discovered at the end of the research.
Another weakness is that, although particle size is one of the main uncertainties, no particle size variation tests were performed for the IFP simulations with multiple particles. The choice to keep the particle size constant was made because the other particle sizes used in the lab tests were less suitable for the simulations as mentioned in the methodology.
The strength of the methodology is that the solver is used with only minimal additions to the models. Even without large changes, the simulations predict the trends in the velocity and the cluster formation processes quite well. Adding more physics to the solver would have taken a lot of time, which was not available since creating the Apptainer container to run the solver initially had already taken a lot of time.

8 Conclusions and Recommendations

The conclusions found in this thesis will be discussed per sub-question. After that, the recommendations will be addressed.

8.1 Conclusions

Do the simulations predict the same cluster formation and flow effects as in model lab tests?

The cluster formations and flow effects predicted by the simulations are similar to those of the model lab tests for IFP angles of 30 to 60 degrees. The simulations show that the cluster formation process also occurs without the random rock sizes and shapes. The fluid velocity field and the resulting drag forces are the main driving forces behind the creation of the clusters.
In the simulations, a distribution with higher and lower particle concentrations is required to start the cluster forming. Evenly distributed particles in the IFP do not form clusters. This could be different in the model lab tests, as the randomly sized and shaped rocks would not stay in an evenly distributed line.

The particles' chaotic movement is missing for the simulations of the IFP at 75 and 90 degrees. This may be attributed to the missing physics in the solver.

To what extent can the solver predict the rock velocity and production for different angles of the inclined fall pipe and different mixture concentrations in the IFP?

The solver does not correctly predict the values for velocity per production. The simulations overpredict the velocity for all angles and particle concentrations. Production is afterward calculated with the velocity and thus also overpredicted. The predictions show the same trends as those in the lab tests, meaning the predicted velocity and production rates can still be used to compare situations or parameters qualitatively.

What is the influence of various system parameters?

The system parameter variation resulted in the understanding that the time step and the grid size were chosen fine enough. Finer values for these parameters influence the solutions of the simulation so little that they can be neglected. Using larger values may introduce errors to the solution.
The last variable that has been varied is the distance factor. Varying this parameter changes a lot in the simulations. Both the particle behavior and the particle velocity are influenced significantly. When using a large value of 3, because of the shear flow, the solver will use fluid velocities that are too much influenced by return flow and not take local differences into account. When using too small values, the solver may not perform correctly anymore, as the kernel uses too few cells to distribute the particle concentration and interaction forces. With such significant differences between two values (1 and 3) used in simulations, it would be interesting to investigate how the simulations might change with more gradual steps.

How can the prediction capability of the solver be improved?

The prediction capability could be improved in one of the following ways. The first and most important improvement to the solver would be the addition of the missing physics. The Magnus lift force, the turbulent dispersion force, forces due to vortex shedding and shape lift force all exert a force perpendicular to the slip velocity. The fact they are not included in the solver is most likely the reason the particle behavior for angles of 75 and 90 degrees is different. The over-prediction of the particle velocity has several possible reasons.
The first option is that the missing forces also slow down the particles in the direction of the pipe. The sideway movements can make particles end up in the return flow, slowing them down. The movement of multiple particles can also cause a mixing effect of the downflow and return flow, slowing both flows and the

particles down. The last effect the sideways movements can have is the earlier breaking apart of rock clusters, which slows the particles down as they move slower alone than together in a cluster.
The second option is that one of the system parameters, the distance factor, has to be calibrated. The result of the simulations is highly dependent on the distance factor value. Incorrect values for the distance factor result in incorrect fluid velocities experienced by the particles. These influence the cluster formations due to local fluid velocity differences and the average particle velocities averaging the fluid velocity from certain parts in the shear flow.
The third option that could enhance particle velocity prediction is including varying rock sizes and shape factors. Either truly different values for each particle or their effect on the C_d value being measured and included. The value of C_{factor} found in Appendix D was not representative for all situations.

To conclude, the main question "To what extent can the rock behavior inside the IFP be predicted using a semi-resolved CFD-DEM method?" can be answered with:

The semi-resolved CFD-DEM solver of Nieuwboer shows promising results for predicting the rock processes inside the inclined fall pipe. The rock cluster forming behavior shows clear similarities with Fontijn's lab tests. The velocities and the velocity calculated production rates are overpredicted but show the same trends as in the lab tests. Several additions and calibrations are proposed to enhance the predicting capabilities.

8.2 Recommendations

The recommendations are structured as answers to a set of questions that arose over the duration of the research.

Would the solver be recommended for further research into IFPs, or are other numerical methods better suited, in hindsight?

The semi-resolved CFD-DEM solver is imperfect at simulating rocks in an inclined fall pipe. With enough time on hand, many simplifications and missing physics could be solved, but the shape of the particles will always remain a sphere or a combination of spheres. Models could be added for drag and lift coefficient values that depend on the orientation, but the fluid flow will always only react to the spheres. The fluid flow has other imperfections because of how the kernel and quadratic functions work with the fluid-solid interactions. When real rock shapes need to be used, and the flow field needs to be studied exactly, the only option is to use a fully resolved method. These methods use many resources and time, making a full IFP simulation very costly.
The semi-resolved method is recommended for all simulations where the flow around the particles is not needed in detail because of its speed and performance, which could even be enhanced, as discussed in the conclusion.

What is the first enhancement that is recommended to be performed on the solver?

With the main goal of improving the results of the solver, the first recommended enhancement to the solver would be the calibration of the distance factor. The distance factor greatly affects the velocity and behavior of the particles and causes a large uncertainty in the solution. Lab data of groups of marbles along a slope, just like in the IFP, but simplified, could be used for this calibration.
Intending to improve the methodology of the experiments performed, the first recommended enhancement to the solver would be the addition of the option to create and destroy particles during the simulation. This option is in the DPM solver of OpenFOAM, on which the solver is based, but it has not been tested properly to work with the kernels of the solver used. Adding this option would allow simulations to continue infinitely with particles inside the test area, giving better results as the process has enough time to become stationary.

What variations on the simulations performed are recommended to enhance the understanding of the cluster formation process?

Without any changes to the solver, the recommendation is to perform the simulations with smaller particles and larger amounts of particles. Simulations will likely take much longer to run, but it should be possible. The interesting part of these simulations is that a thicker layer of randomly placed particles has less obvious gaps between the particles from which the clusters can start to form, but there are higher and lower concentrated areas in the layer. Knowing if these areas are enough to start the cluster formations would be interesting.
With small changes to the code, the recommendation would be to perform a simulation with a couple of different sizes of particles; 3 sizes would probably suffice. A change in the interaction distance between particles is probably necessary so this setting can differ per particle diameter. This change in simulation can show the effect of the different-sized particles on the cluster formations. The larger particles should move faster than the smaller particles, both before the particles form clusters and after. The hypothesis is that the clusters start forming faster but also fall apart faster due to the size differences. It would be interesting to compare those results to the results from this report and tell if this hypothesis is correct.

Which recommendations can be given to enhance the experimental data for better use as simulation validation data?

More information about the used rocks would reduce uncertainties a lot. A gradation is quite broad; the rocks could all be of a size in the upper half or lower half and still be part of the gradation. The total mass of the particles, in combination with a count of the particles, gives a better representation of the average volume of the particles.
For the best validation data, the lab tests should mimic a couple of the solver's restrictions. A simulation with many marbles, all introduced to the pipe at once as a kind of dam break test, is perfectly reproducible for the solver and could help find correct values for the distance factor or some other uncertain parameter.

With the results found, what would the recommendation be to limit the cluster formation in operations?

The results show that the particles do not form clusters when they start in a line with the same distance from each other. Creating this in operations is impossible, as the particles can not start over the entire length of the pipe; in operations, the particles all have to start at the top of the pipe. Furthermore, the different sizes and shapes of the particles will cause the line to break up too.
The results concluded that the differences in the fluid velocity field and the following drag forces are the main driving forces behind the cluster formation. This means that reducing the slip velocity between the particles and fluid would reduce the driving force for the cluster formations. This could be achieved in multiple ways. Pumping water in the top of the pipe and creating a hole below the water surface are two options for trying to create a net flow down the pipe. First, the particles will move faster down with only a little water added. However, with more water, it can be reasoned that the particle-wall friction will start to slow the particles down, and the slip velocity between the water and the particles will drop. The exact behavior is, however, unknown, as is the volume of water required.
Other options for lowering the slip velocity could be widening the pipe or using an oval-shaped pipe, as the flow up would be further away from the flow down, and the shear rate above the particle layer could be lower. However, it is possible that both flows become stronger, and the shear rate and slip velocity for the particles stay the same. If this is the case, actively heightening the wall friction could slow the flows down. Unfortunately, this does increase the wear on the material.

A Apptainer

This Appendix discusses the main functions of Apptainer. Afterwards, it discusses the creation and use of the Apptainer container containing Ubuntu, OpenFOAM, and the OpenFOAM solver.

The exact versions of software used are:

- Ubuntu 18.06
- OpenFOAM-v1712
- BigGaussDPMFoam

A.1 Main functionality

A.1.1 Def file

From: https://apptainer.org/docs/user/main/definition_files.html

Creating a file system, installing OpenFOAM and packing it to a sif file, the "old way" gave some errors during the packing. The new way is to do this by using a def file. This section will talk about (the use of) the def file.

An Apptainer Definition File (or "def file" for short) is like a set of blueprints explaining how to build a custom container. It includes specifics about the base OS to build or the base container to start from, software to install, environment variables to set at runtime, files to add from the host system, and container metadata.

An Apptainer Definition file is divided into two parts:

- Header: The Header describes the core operating system to build within the container. Here you will configure the base operating system features needed within the container. You can specify the Linux distribution, the specific version, and the packages that must be part of the core install (borrowed from the host system).
- Sections: The rest of the definition comprises sections (sometimes called scriptlets or blobs of data). Each section is defined by a % character followed by the name of the particular section. All sections are optional, and a def file may contain more than one instance of a given section. Sections executed at build time are executed with the /bin/sh interpreter and can accept /bin/sh options. Similarly, sections that produce scripts to be executed at runtime can accept options intended for /bin/sh.

Header

The header should be located at the beginning of the def file. It tells Apptainer about the base operating system used to build the container. It is composed of several keywords.

The only keyword that is required for every type of build is Bootstrap. It determines the bootstrap agent that will be used to create the base operating system you want to use.

Each bootstrap agent enables its own options and keywords, which you can read about, and see examples of, in the appendix. https://apptainer.org/docs/user/main/appendix.html#buildmodules

The Bootstrap keyword needs to be the first entry in the header section. Other keywords may also be valid in the header depending on the value assigned to Bootstrap. When using docker, the From keyword is mandatory. Basic layout:

- Bootstrap: docker
- From: <registry>/<namespace>/<container>:tag>@<digest>

Other keywords are optional. These are:

- registry. defaults to index.docker.io.
- namespace. defaults to library. This is the correct namespace for some official containers (ubuntu for example).
- tag. defaults to latest

Here is an example that uses the From keyword to build a Debian container:

- Bootstrap: docker
- From: debian:7

For the research, Ubuntu 18.04 is used. This can be pulled from docker. Oversight of versions is given on https://hub.docker.com/_/ubuntu.

As described above, the header becomes:

- Bootstrap: docker
- From: ubuntu:bionic

Sections

After pulling the main image with the header, different parts of the build are broken into sections. Different sections add different content or execute commands at different times during the build process. Note that if any command fails, the build process will halt. The order of the sections in the def file is unimportant, they have a set order of execution. The available sections in the order of execution:

- arguments
- setup
- files
- app
- post
- test
- environments
- startscript
- runscript
- labels
- help

%arguments Arguments can be used to set values for set entry points somewhere else in the def file. Example:

- Bootstrap: docker
- From: ubuntu: VERSION
- Stage: build
-
- %arguments
- VERSION=22.04

The original value can also be overwritten while building with –build-arg or -build-arg-file.

- apptainer build –build-arg VERSION=23.04 my_container.sif my_container.def

%setup During the build process, commands in the %setup section are first executed on the host system outside of the container, after the base OS has been installed. You can reference the container file system with the $APPTAINER_ROOTFS environment variable in the %setup section. More recent versions of Apptainer provide the %files section, which is a safer alternative to copying files from the host system into the container during the build process.

%files The %files section allows you to copy files into the container with greater safety than using the %setup section. Its general form is:

- %files [from <stage>]
- <source> [<destination>]
- ...

%app The %app section is for when multiple versions of an app have to run on the same/comparable environment. Then it is possible to create sub-app versions within one apptainer environment. This is not necessary in this instance.

%post This section is where you can download files from the internet with tools like git and wget, install new software and libraries, write configuration files, create new directories, update the system, etc.
Example:

- %post
- apt-get update && apt-get install -y netcat
- NOW=‘date‘
- echo ”export NOW=” $NOW” ” >> $APPTAINER_ENVIRONMENT

Besides installing netcat, the example script also sets an environment variable at build time. Note that the value of this variable cannot be anticipated, and therefore cannot be set earlier in the %environment section. For situations like this, the $APPTAINER_ENVIRONMENT variable is provided. Assigning a value to this variable will cause it to be written to a file called /.singularity.d/env/91-environment.sh that will be sourced by the container at runtime.

%test The %test section runs at the very end of the build process, and can be used to validate the container using methods of your choosing. You can also execute this scriptlet through the container itself, using the test command.

%environments The %environment section allows you to define environment variables that will be set at runtime. Note that these variables are made available in the container at runtime, but not at build time. This means that if you need the same variables during the build process, you should also define them in your %post section. Specifically:

during build: The %environment section is written to a dedicated file in the container metadata directory. This file is not sourced.

during runtime: The file in the container metadata directory is sourced.

You should use the same conventions that you would use in a .bashrc or .profile file.

%startscript Similar to the %runscript section, the contents of the %startscript section are written to a dedicated file within the container at build time. This file is executed at the instance the start command is issued.

%runscript The contents of the %runscript section are written to a dedicated file within the container that is executed when the container image is run (either via the apptainer run command or by executing the container directly as a command). When the container is invoked, arguments following the container name are passed to the runscript. It is heavily advised to use process arguments within the runscript.

%labels The %labels section is used to add metadata to the file /.singularity.d/labels.json within your container. The general format is a name-value pair.

%help Any text in the %help section is transcribed into a dedicated metadata file in the container during the build process. This text can then be displayed using the run-help command.

A.2 Creation of the container

A.2.1 Building sections

Steps that need to be taken:
——File———

- Create an OpenFOAM folder at -/opt/OpenFOAM
- Unpack OpenFOAM tgz files in folder
- Unpack solver bas into temp folder
  —-Environments—-
- Add LC_ALL=C
  —-Post———
- Run apt-get update
- Install dependencies OpenFOAM: apt-get install git-core build-essential cmake .....etc.
- Source the openfoam bashrc file
- Install third party
  - cd third party directory
  - export qt select
  - make paraview
  - refresh wmRefresh
- Install openFOAM
  - cd WM project
  - export qt_select
  - allwmake
- Move Bas' solver from temp into run folder
- compile solver v1 and v2
  —-Test——
- Test

A.2.2 Used def file

The following def file is used:

```
Bootstrap: docker
From: ubuntu:bionic

%files
/scratch/mhverhoog/Apptainer/Sources/OpenFOAM-v1712.tgz /opt/OpenFOAM/
/scratch/mhverhoog/Apptainer/Sources/ThirdParty-v1712.tgz /opt/OpenFOAM/
/scratch/mhverhoog/Apptainer/Sources/bigGaussDPMFoam1712.zip /opt/OpenFOAM/
/scratch/mhverhoog/Apptainer/Sources/bigGaussDPMFoam1712CdVar.zip /opt/OpenFOAM/
/scratch/mhverhoog/Apptainer/Sources/bashrc_edit /opt/OpenFOAM/bashrc


%environment
export LC_ALL=C

%post
echo "update and install dependencies"
apt-get -y update
apt-get -y install git-core build-essential cmake libfl-dev bison zlib1g-dev qttools5-dev qtbase5-dev
libqt5x11extras5-dev gnuplot libreadline-dev libncurses-dev libxt-dev libopenmpi-dev openmpi-bin libboost-
system-dev libboost-thread-dev libgmp-dev libmpfr-dev python python-dev libcgal-dev curl
apt-get -y install libglu1-mesa-dev
apt-get -y install time unzip


echo "unpack third party & OpenFOAM"
tar zxvf /opt/OpenFOAM/OpenFOAM-v1712.tgz -C /opt/OpenFOAM/
tar zxvf /opt/OpenFOAM/ThirdParty-v1712.tgz -C /opt/OpenFOAM/


rm /opt/OpenFOAM/OpenFOAM-v1712.tgz
rm /opt/OpenFOAM/ThirdParty-v1712.tgz


#!/bin/bash

echo "source bashrc file"
mv /opt/OpenFOAM/bashrc /opt/OpenFOAM/OpenFOAM-v1712/etc/bashrc
. /opt/OpenFOAM/OpenFOAM-v1712/etc/bashrc FOAMY_HEX_MESH=yes

echo "install third party"
cd $WM_THIRD_PARTY_DIR
export QT_SELECT=qt5
./makeParaView -python -mpi -python-lib /usr/lib/x86_64-linux-gnu/libpython2.7.so.1.0

echo "refresh wmRefresh"
. /opt/OpenFOAM/OpenFOAM-v1712/etc/bashrc $FOAM_SETTINGS

echo "install OpenFOAM"
cd $WM_PROJECT_DIR
export QT_SELECT=qt5

./Allwmake

echo "refresh wmRefresh"
# . $WM_PROJECT_DIR/etc/config.sh/unset (does not work)
. /opt/OpenFOAM/OpenFOAM-v1712/etc/bashrc $FOAM_SETTINGS

echo "unpack solver"
unzip /opt/OpenFOAM/bigGaussDPMFoam1712.zip -d $WM_PROJECT_USER_DIR
rm /opt/OpenFOAM/bigGaussDPMFoam1712.zip

echo "unpack solver2"
unzip /opt/OpenFOAM/bigGaussDPMFoam1712CdVar.zip -d $WM_PROJECT_USER_DIR
rm /opt/OpenFOAM/bigGaussDPMFoam1712CdVar.zip
```

```
echo "Compile solver"
cd $WM_PROJECT_USER_DIR/bigGaussDPMFoam1712
./Allwmake

echo "Compile solver2"
cd $WM_PROJECT_USER_DIR/bigGaussDPMFoam1712CdVar
./Allwmake

%runscript
. /opt/OpenFOAM/OpenFOAM-v1712/etc/bashrc $FOAM_SETTINGS
exec $@

%startscript
#

%test
. /opt/OpenFOAM/OpenFOAM-v1712/etc/bashrc
icoFoam -help
bigGaussDPMFoam -help
```

B Changes made to the BigGaussDPMFoam solver

Over the course of this thesis, two changes have been made to the solver. The first change was made because the interaction force from the particles back to the fluid was not calculated correctly. The second change was made to incorporate an extra multiplication factor into the particle drag formulation. In order to be able to add a factor to the C_d value as described in section 5.4. The exact changes to the code will be shown in this appendix.

B.1 Interaction force fix

When using the BigGaussDPMFoam solver with only one processor and a single grid, an issue used to occur with the calculation of the interaction forces of the particle to the mesh.
This issue originates from the DistributedKinematicCloud.C file, found in the folder: /bigGaussDPMFoam/langrangianintermediate/clouds/Templated/DistributedKinematicCloud/.
The issue is caused by too many comments in the postEvolve function. In the 2019 version of the solver the part of the code looks like this:

```
template<class CloudType>
template<class TrackCloudType>
void Foam::DistributedKinematicCloud<CloudType>::postEvolve
(
    TrackCloudType& cloud,
    typename parcelType::trackingData& td
)
{
        CloudType::postEvolve();
        Info << "Distributing Particle data on mesh" << endl;
        cloud.spreadParticles(cloud, td, false); //zero states that no particles are added to the
            cloud in the postEvolve fase.
/*        forAllIter(typename DistributedKinematicCloud<CloudType>, *this, pIter) // Loop over all
            particles
        {
                parcelType& p = pIter();
                p.getWeights();
                p.scaleWeights(); // scale the weights of the particles that are
                not on a processor patch (to be summed 1)
                p.divideSourceTerms(cloud); //distribute the Ucoef and Utrans per
                particle over the whole field.
        }
        //compute the particle concentration per volume cell
        compute_theta();
        //spreadParticles(cloud,td);
*/
}
```

The issue was solved by removing the comments for the loop over all particles and the computing of the theta. The scaling of the forces to the weights of the particles is not necessary here and can stay commented. The same part of code after the fixes looks like this:

```
template<class CloudType>
template<class TrackCloudType>
void Foam::DistributedKinematicCloud<CloudType>::postEvolve
```

```
(
    TrackCloudType& cloud,
    typename parcelType::trackingData& td
)
{
        CloudType::postEvolve();
        Info << "Distributing Particle data on mesh" << endl;
        cloud.spreadParticles(cloud, td, false); //zero states that no particles are added to the
            //cloud in the postEvolve fase.
        forAllIter(typename DistributedKinematicCloud<CloudType>, *this, pIter) // Loop over all
            //particles
        {
                parcelType& p = pIter();
                //p.getWeights();
                //p.scaleWeights(); // scale the weights of the particles that are not on a processor
                //patch (to be summed 1)
                p.divideSourceTerms(cloud); //distribute the Ucoef and Utrans per particle over the
                //whole field.
        }
        //compute the particle concentration per volume cell
        compute_theta();
        //spreadParticles(cloud,td);

}
```

B.2 Addition of the C_{factor}

For the addition of a changeable parameter in the solver, a couple of files had to be changed. First, the files with which the drag is calculated. These are the files of the diFeliceDrag formulation, as this drag function includes the hindered settling formulation. The files deFeliceDrag.C and deFeliceDrag.H are in the folder /bigGaussDPMFoam/lagrangianintermediate/kinematicSubModels/ParticleForces/deFeliceDrag/. To ensure that the original deFeliceDrag is also usable in the new solver, a new folder named deFeliceDragFactor contains a copy of the files, also with the ...Factor added to their name.
A file that did change, but is not shown in the appendix is the makeBasicDistributedKinematicCollidingParcelSubmodels.C in folder /bigGaussDPMFoam/lagrangianIntermediate/parcels/derived /basicDistributedKinematicCollidingParcel/.This file contains a list of all files for drag models and other particle force files included in the solver. The new diFeliceDragFactor.H file is added to this list.

Differences besides the name change between the deFeliceDrag.C and deFeliceDragFactor files are an addition in the constructors for reading the C_{factor} value named Factor_ and the calculation of the force on the particle as shown below.

The original constructors.

```
template<class CloudType>
Foam::diFeliceDrag<CloudType>::diFeliceDrag
(
    CloudType& owner,
    const fvMesh& mesh,
    const dictionary& dict
)
:
    ParticleForce<CloudType>(owner, mesh, dict, typeName, false)
{}
```

The new constructors

```
template<class CloudType>
Foam::diFeliceDragFactor<CloudType>::diFeliceDragFactor
(
    CloudType& owner,
```

```
    const fvMesh& mesh,
    const dictionary& dict,
    const word& forceType

)
:
    ParticleForce<CloudType>(owner, mesh, dict, forceType, true),
    Factor_(readScalar(this->coeffs().lookup("CdFactor")))
{
Info<< "Constructor for diFeliceDragFactor called with Factor_: " << Factor_ << endl;
}
```

The original force calculation

```
template<class CloudType>
Foam::forceSuSp Foam::diFeliceDrag<CloudType>::calcCoupled
(
    const typename CloudType::parcelType& p,
    const typename CloudType::parcelType::trackingData& td,
    const scalar dt,
    const scalar mass,
    const scalar Re,
    const scalar muc
) const
{

   //scalar alphac(alphac_[p.cell()]);

        scalar alphac =  p.alphaAtParticle();
        alphac=max(alphac,0.01); // the alphac has a minimum of 0.01. This is also the default
            //value in the kinematic properties dict. It would be better to read that one here.
        alphac=min(alphac,1.0); // the maximum value is 1. After initializing the value can reach
            //higher values.


    forceSuSp value = forceSuSp
    (
        Zero,
        (mass/p.rho())
       *0.75*CdRe(alphac*Re)*muc*pow(alphac, - Beta(Re))/sqr(p.d())
    );

        return value;
}
```

The new force calculation

```
template<class CloudType>
Foam::forceSuSp Foam::diFeliceDragFactor<CloudType>::calcCoupled
(
    const typename CloudType::parcelType& p,
    const typename CloudType::parcelType::trackingData& td,
    const scalar dt,
    const scalar mass,
    const scalar Re,
    const scalar muc
) const
{

   //scalar alphac(alphac_[p.cell()]);

        scalar alphac =  p.alphaAtParticle();
        alphac=max(alphac,0.01); // the alphac has a minimum of 0.01. This is also the default
```

```
            //value in the kinematic properties dict. It would be better to read that one here.
        alphac=min(alphac,1.0); // the maximum value is 1. After initializing the value can reach
            //higher values.

//Add the Cd factor at the calculation on the force on the particle
    forceSuSp value = forceSuSp
    (
        Zero,
        (mass/p.rho())
       *Factor_*0.75*CdRe(alphac*Re)*muc*pow(alphac, - Beta(Re))/sqr(p.d())
    );

        return value;
}
```

The diFeliceDrag.H file also needed to be changed. Here, the Factor_ scalar needed to be declared, and the declaration of the constructors needed an extra function to be able to look up the value of the C_{factor}, $Factor\_$.

The original declarations

```
template<class CloudType>
class diFeliceDrag
:
    public ParticleForce<CloudType>
{
    // Private Data

    // Private Member Functions

        //- Drag coefficient multiplied by Reynolds number
        inline scalar CdRe(const scalar Re) const;

        //- Beta factor based on Di Felice 1994
        inline scalar Beta(const scalar Re) const;


            // - index of celli in thetacells (for determining the concentration of this
            //particle in cell i)
            label thetaCelliIndex
            (
                    const labelList& thetaCells,
                    const label celli
            ) const;

public:

    //- Runtime type information
    TypeName("diFeliceDrag");


    // Constructors

        //- Construct from mesh
        diFeliceDrag
        (
            CloudType& owner,
            const fvMesh& mesh,
            const dictionary& dict
        );

        //- Construct copy
        diFeliceDrag(const diFeliceDrag<CloudType>& df);
```

```
//- Construct and return a clone
virtual autoPtr<ParticleForce<CloudType>> clone() const
{
    return autoPtr<ParticleForce<CloudType>>
    (
        new diFeliceDrag<CloudType>(*this)
    );
}
```

The new declarations

```
template<class CloudType>
class diFeliceDragFactor
:
    public ParticleForce<CloudType>
{
    // Private Data
    scalar Factor_;

    // Private Member Functions

        //- Drag coefficient multiplied by Reynolds number
        inline scalar CdRe(const scalar Re) const;

        //- Beta factor based on Di Felice 1994
        inline scalar Beta(const scalar Re) const;


                // - index of celli in thetacells (for determining the concentration of this
            //particle in cell i)
                label thetaCelliIndex
                (
                        const labelList& thetaCells,
                        const label celli
                ) const;

public:

    //- Runtime type information
    TypeName("diFeliceDragFactor");


    // Constructors

        //- Construct from mesh
        diFeliceDragFactor
        (
            CloudType& owner,
            const fvMesh& mesh,
            const dictionary& dict,
            const word& forceType = typeName
        );

        //- Construct copy
        diFeliceDragFactor(const diFeliceDragFactor<CloudType>& df);

        //- Construct and return a clone
        virtual autoPtr<ParticleForce<CloudType>> clone() const
        {
            return autoPtr<ParticleForce<CloudType>>
            (
                new diFeliceDragFactor<CloudType>(*this)
            );
        }
```

C Studies on the simulation times and visible effects for different grid and particle parameters.

Multiple 2D and 3D simulations have been performed to gain a better understanding of the IFP simulations' duration and necessary particle volumes for effects beyond the startup phase to appear.

C.1 Domain and calculation grid

Domain

The 2D and 3D domains consist of a rectangular in 2D and block-shaped in 3D (from now on just referred to as block-shaped) flow field and a flow field in a pipe. The pipe starts outside the flow field and partly overlaps with the block. The end of the pipe is around half the field. An illustration of this can be seen in Figure C.1. The stones will be placed at the top of the pipe and fall through the pipe exit into the block-shaped field.

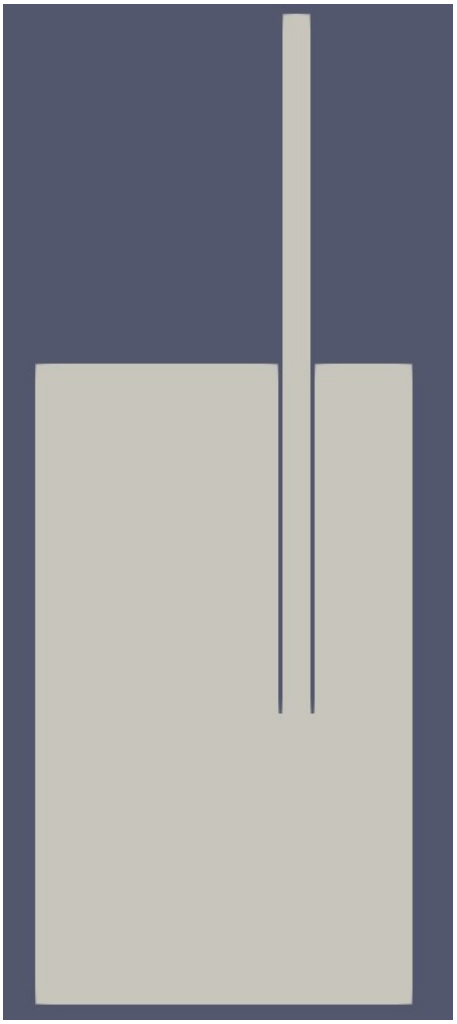


Figure C.1: The domain, consisting of the inside of a pipe and a rectangular flow field.

2D grid

In Figure C.2, the 2D grid can be seen. The domain has a thickness of 1 cell in the y-direction, as seen in Figure C.3, corresponding to the used rock size (10mm or 20mm). The cells all have the same size: 10 cm by 10 cm in the x and z directions. As shown in Figure C.4, the only cells slightly deviating from these sizes are the cells on the end of the pipe. The reason for this deviation is the pipe wall thickness of 15mm in the simulations. The grid contains 42624 cells.

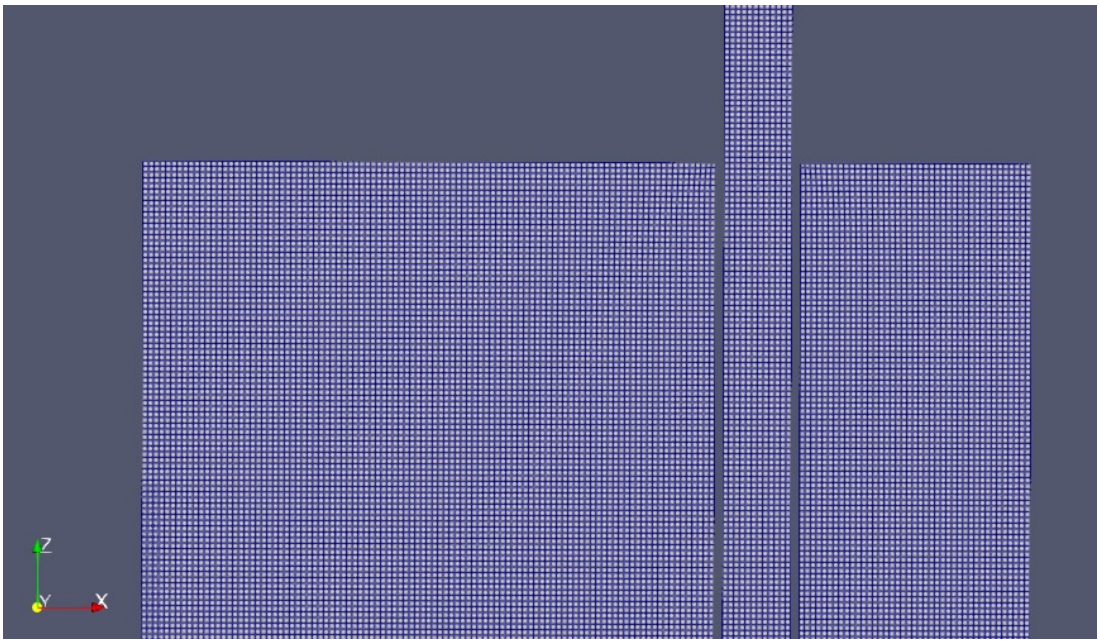


Figure C.2: The (almost uniform) 2D grid.

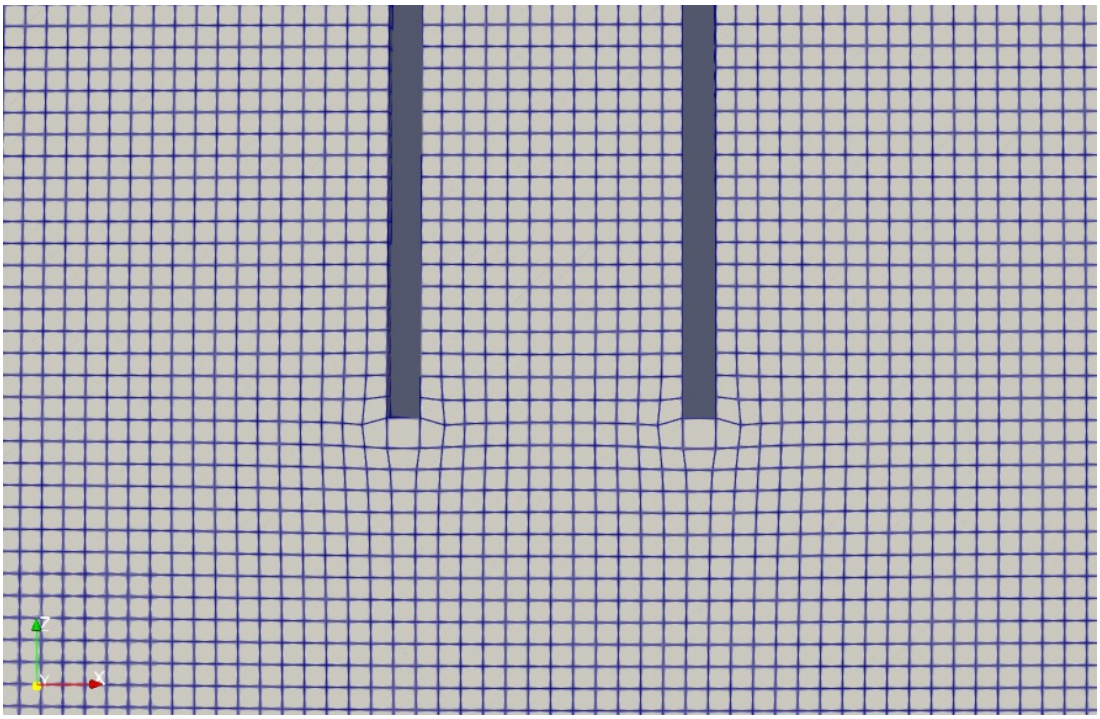


Figure C.3: The cells near the exit deviate from the uniform grid due to the pipe wall thickness.

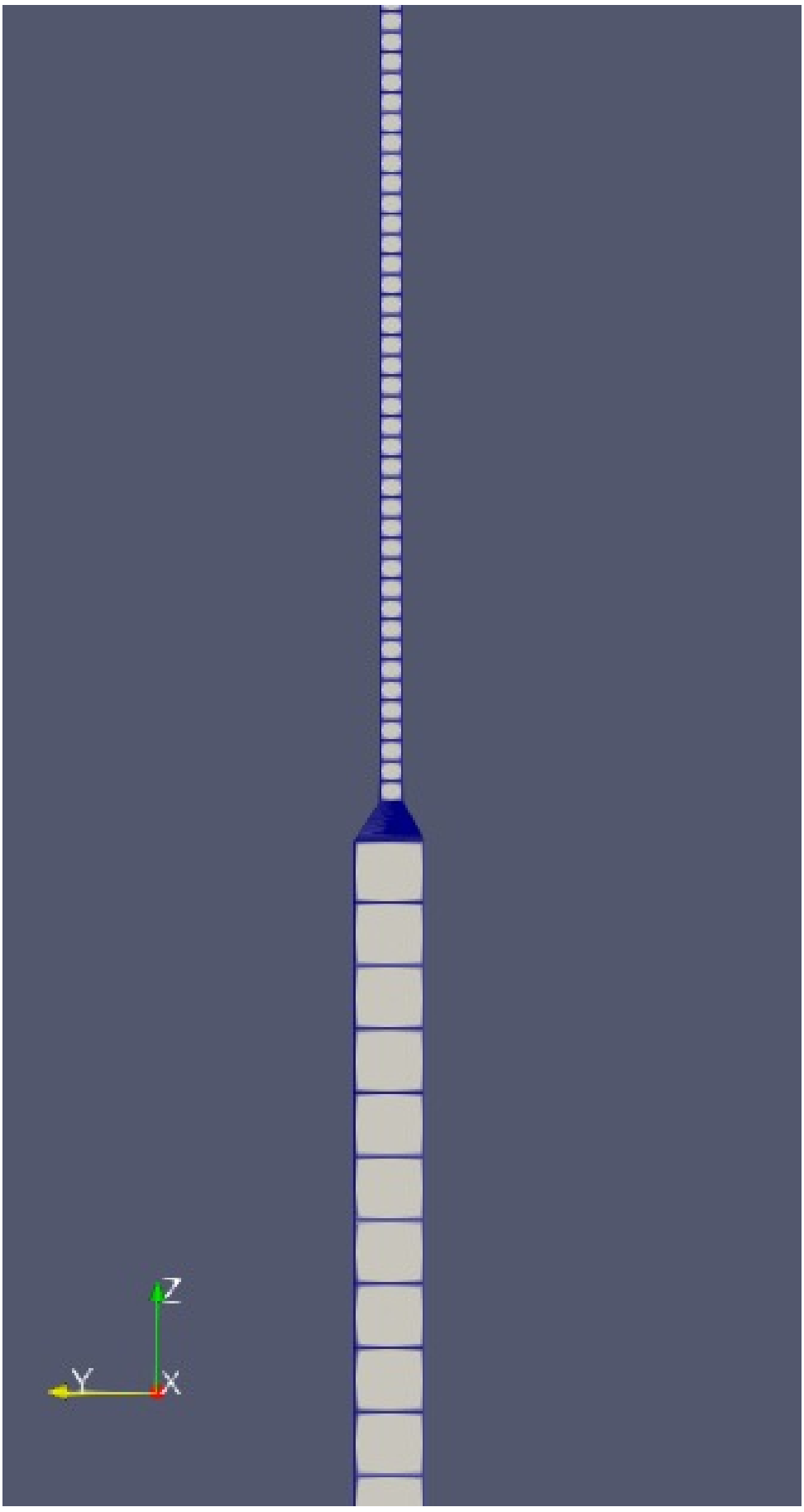


Figure C.4: The side view of the 2D grid.

3D grid

In Figure C.5, the 3D grid can be seen with details in Figures C.6, C.7, C.8, and C.9. The field in the pipe has a cylinder shape. The cells inside the cylinder are the smallest. These are around 3.25 mm in all dimensions. The field below the pipe has an average cell size of 15 mm in all dimensions, and the cells above the end of the pipe are about 60 mm in all dimensions. The pipe's finer grid is used to describe the cylinder shape. The grid below the pipe is the optimal size of about 1 to 1 for the stone sizes, and the grid above the pipe end has larger cells since no stones will reach over there. The grid was generated using snappyHexMesh. Some irregular patterns can be seen near the exit of the pipe and in the top just outside of the pipe. These patterns may cause issues in the simulation. The total number of cells is 2252112.

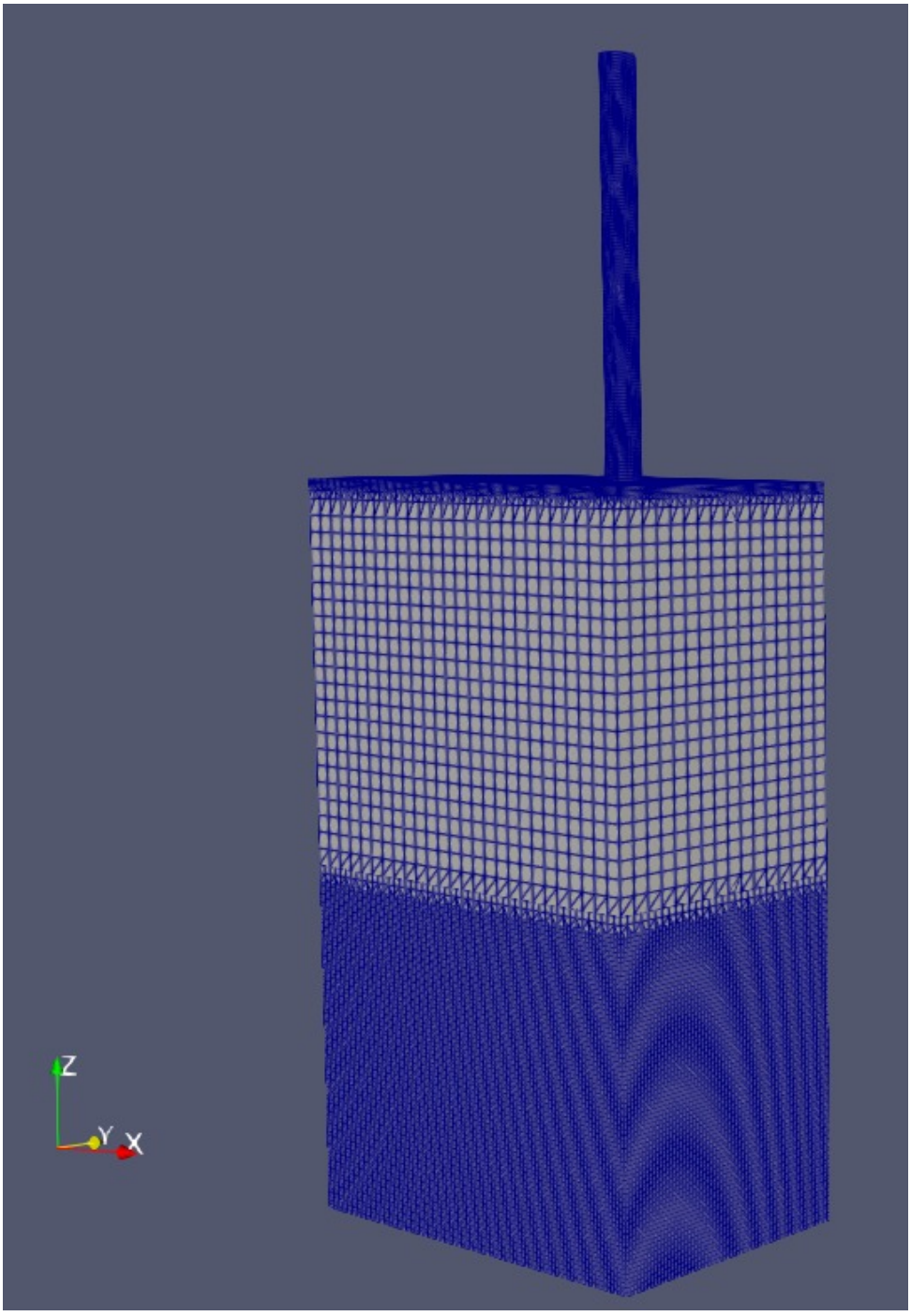


Figure C.5: The 3D grid.

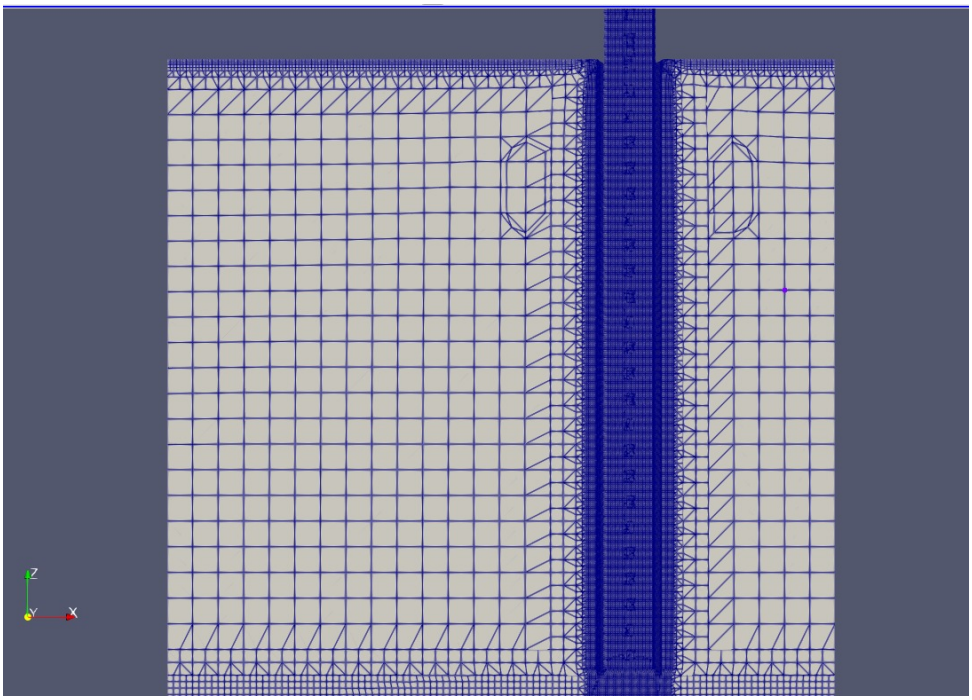


Figure C.6: A slice halfway the 3D grid. Closeup of the pipe section and the area around it.

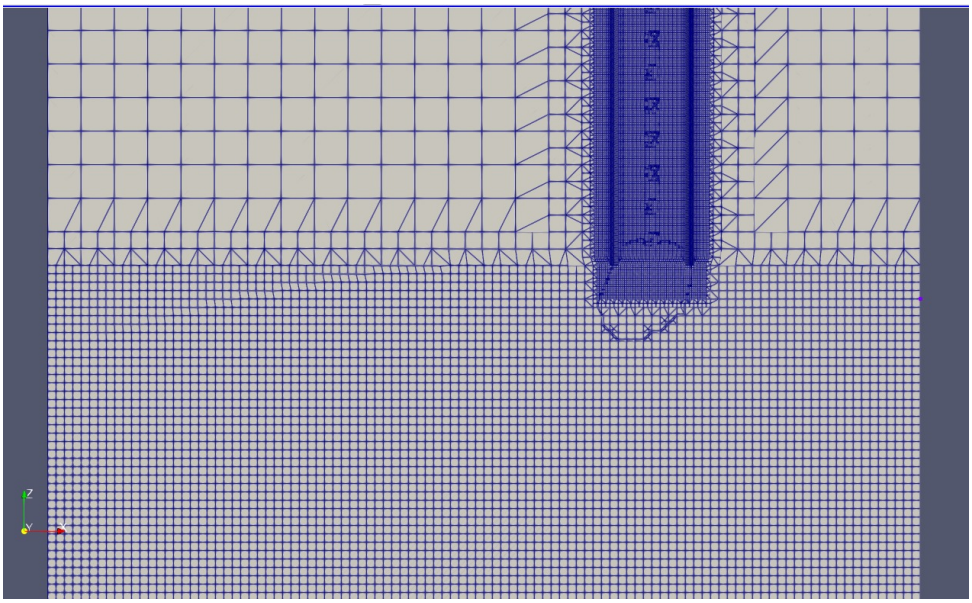


Figure C.7: A slice halfway the 3D grid. Closeup of the section below the pipe exit.

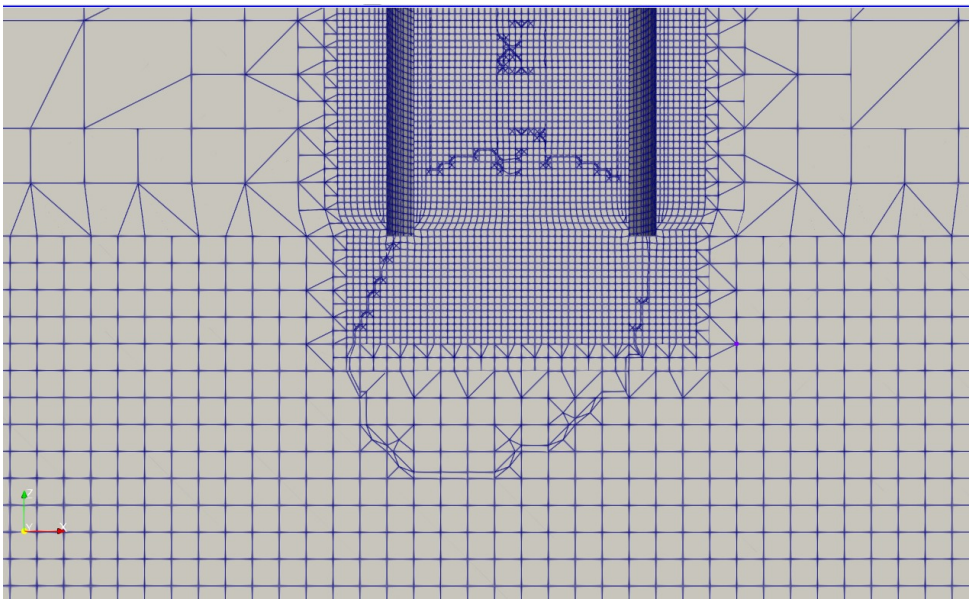


Figure C.8: A slice halfway the 3D grid. Closeup of the pipe exit.

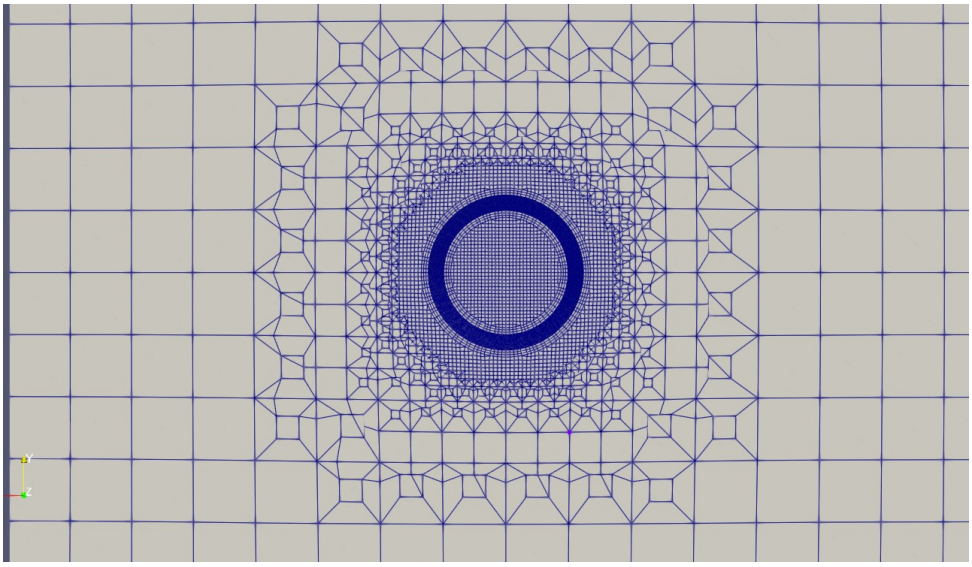


Figure C.9: A slice of the 3D grid. Sliced as a cross-section of the pipe.

3D grid number 2

The calculation times for the first 3D grid are higher than expected. The simulation times can be seen in the next chapter (table C.1). To tackle these large simulation times, the grid has been made a bit coarser in and around the cylinder. The cell size after this change is around 7.5 mm in all directions. The cell size away from the pipe is still the same as in the previous grid, as can be seen in figure C.10, figure C.11, figure C.12, and figure C.13. This version of the grid has also been generated by snappyHexMesh. Some imperfections were still created during the generation, especially in the area outside the pipe and up from the pipe exit. Later in the report, an investigation is done into the question of whether the area above the pipe exit can be shortened.

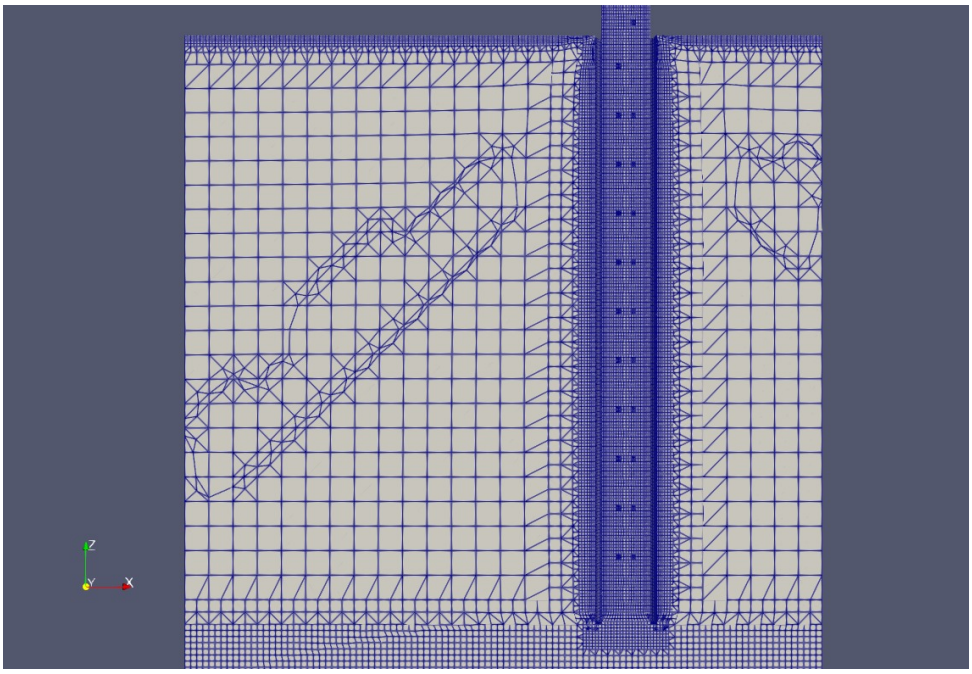


Figure C.10: A slice halfway through the coarser 3D grid. Closeup of the pipe section and the area around it.

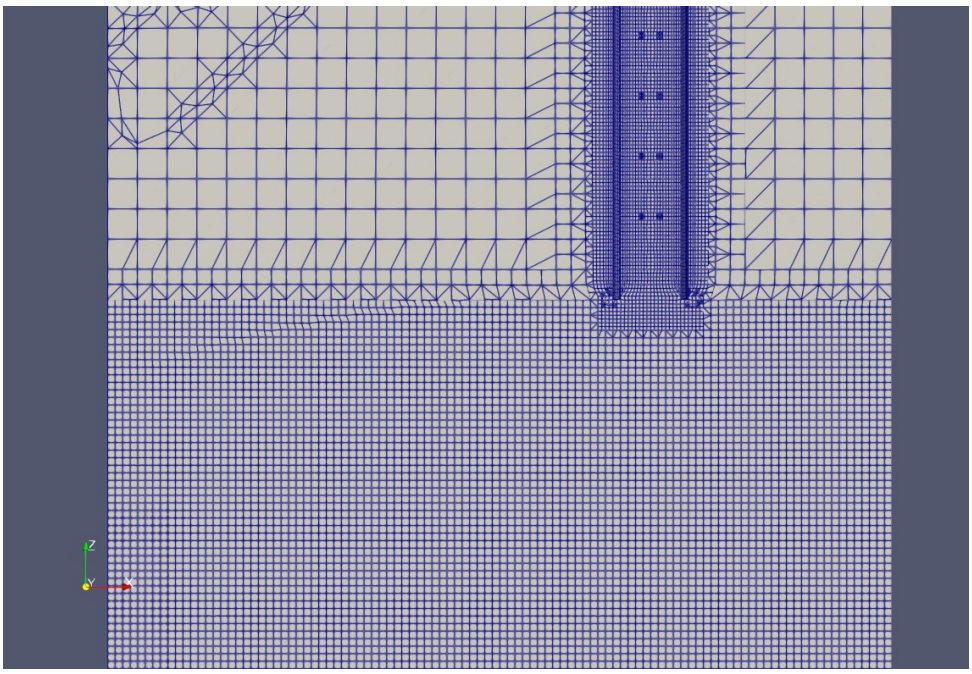


Figure C.11: A slice halfway through the coarser 3D grid. Closeup of the section below the pipe exit.

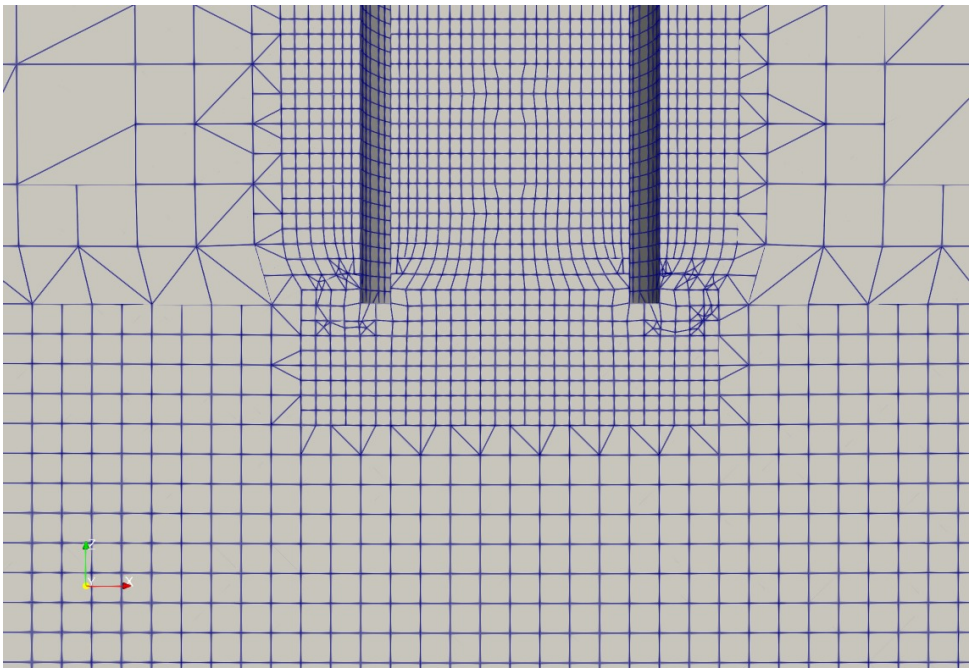


Figure C.12: A slice halfway through the coarser 3D grid. Closeup of the pipe exit.

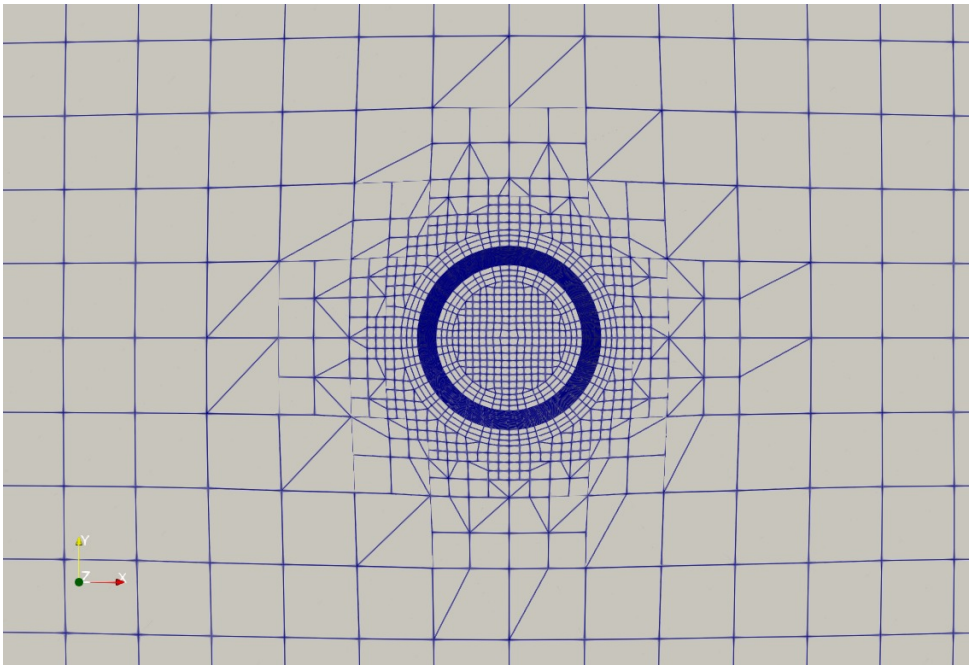


Figure C.13: A slice of the coarser 3D grid. Sliced as a cross-section of the pipe.

3D grid number 3

A third version of the 3D grid (course2_3D) was made. The reasons were the distortions in the velocity field due to the imperfections in the field, as can be seen in C.10, and the realization from flowline data that the block-shaped part of the domain does not need to reach so high above the pipe exit, as will be seen in figure C.35. The grid is kept as homogeneous as possible to combat the generation of imperfections in the grid. When steps from a finer to a coarser part of the grid were necessary, they were done in at least two cells per step instead of one cell before. The entire field of the block-shaped area has the same grid size, except for just around the pipe. The cells are 30mm in all dimensions. This results in a bit larger than the optimal particle cell size ratio of 1:1, but the area below the pipe is a less critical research area. The inside of the pipe has the same cell size as the second 3D grid, 7.5mm in all dimensions. In Figure C.15, it can be seen that in this version, no snapping layers were created around the outside of the pipe. The grid was just as the previous versions generated with snappyHexMesh, but no large areas with imperfections were generated due to the low amount of cell-size steps. The number of cells in the grid is reduced to 312152 cells.

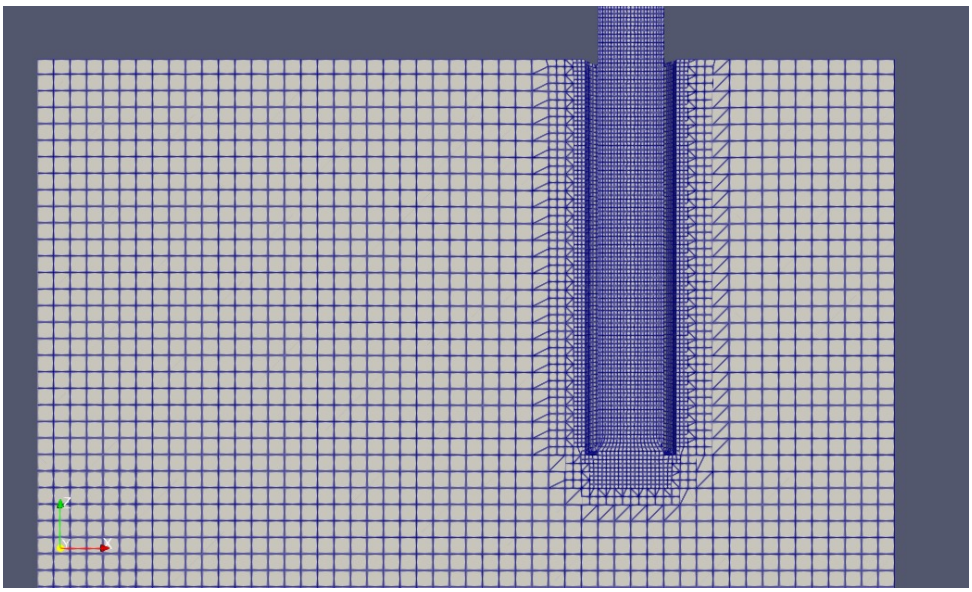


Figure C.14: The third version of the 3D grid has a more uniform grid outside the pipe and larger steps between different cell sizes.

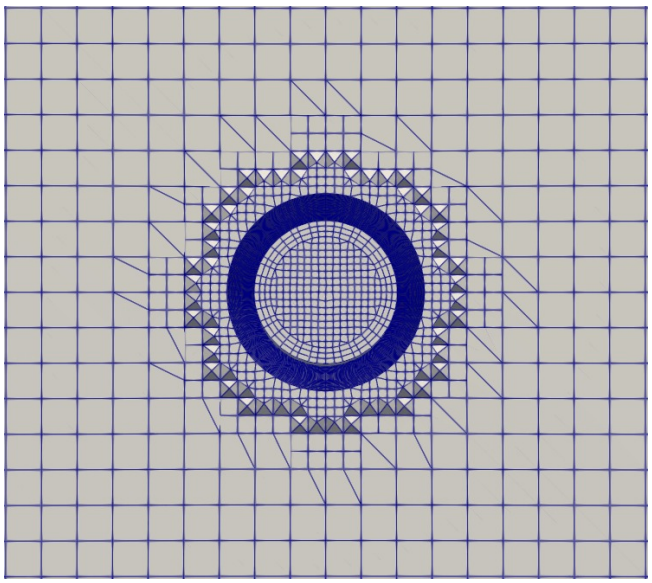


Figure C.15: A slice of the third version of the 3D grid along the cross-section of the pipe.

C.2 Simulations for testing the grids and time estimations

C.2.1 List of simulations and simulation times

Simulation name	Grid	Cells	Angle	Rocks	Simulated time (s)	Calculation time (s)	Calculation time (hours)
IFP2D_30	2D	42624	30	89	10	4106	1.14
IFP2D_45	2D	42624	45	89	10	4014	1.12
IFP2D_60	2D	42624	60	89	10	3780	1.05
IFP2D_75	2D	42624	75	89	10	4098	1.14
IFP2D_90	2D	42624	90	89	10	4353	1.21
IFP2D_MS30	2D	42624	30	178	7	2039	0.57
IFP2D_MS75	2D	42624	75	178	7	2010	0.56
IFP2D_LD75	2D	42624	75	52	6	2089	0.58
IFP2D_RP75	2D	42624	75	239	6	1530	0.43
IFP3D_1	Fine3D	2252112	30	1	10	45080	12.5
IFP3D_98	Fine3D	2252112	30	73	5	381189	105.9
IFP3D_1_2	Coarse_3D	994230	30	1	6.34	31482	8.75
IFP3D_98_2	Coarse_3D	994230	30	73	10	173277	48.13
IFP3D_1_3	Coarse2_3D	312152	75	1	10	40967	11.38
IFP3D_98_3	Coarse2_3D	312152	30	73	10	48329	13.42

Table C.1: List of simulations with their corresponding settings and calculation times.

C.2.2 2D simulations

2D simulations have been performed for five different angles. From 30 degrees up to 90 degrees in steps of 15 degrees. These angles are all compared to a horizontal line. The first five simulations showed the startup phase quite well but lacked the volume of particles for the effects after the startup phase.

The 30-degree simulation, figure C.16, shows a line of rolling particles. The stones in front collided but did not have enough velocity difference to roll over or under the other particles. The rocks in the back half of the group are spread further apart. The last rocks are again closer together.

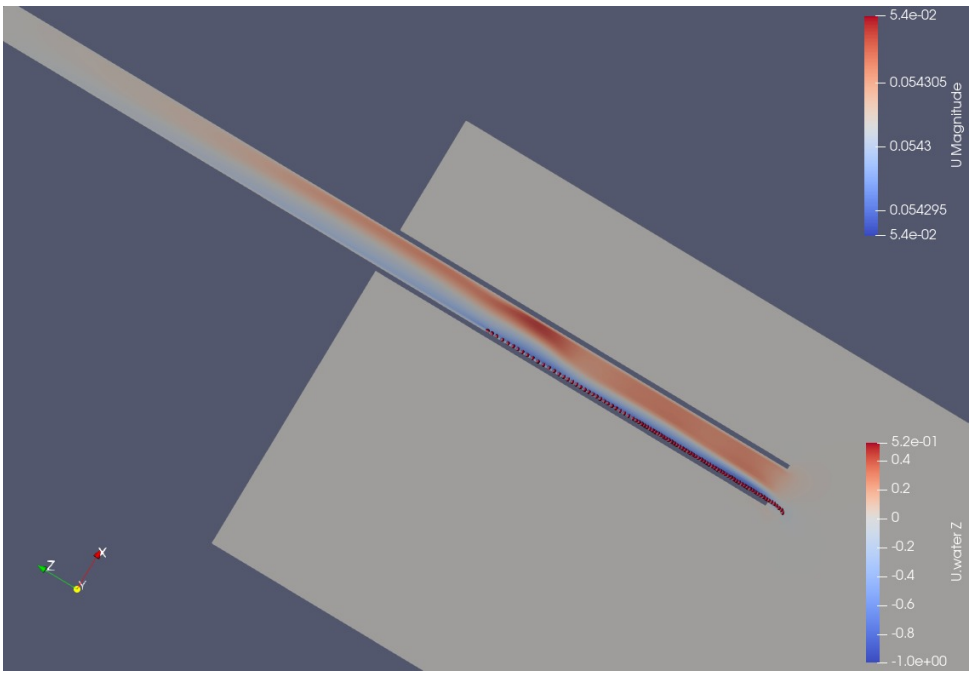


Figure C.16: Rocks in a 2D IFP simulation. With the IFP at an angle of 30 degrees.

In the 45-degree simulation, figure C.18, the rocks in front of the group form a clump. Just behind this clump, a gap appears. After this gap, the rocks are again closer together, further apart, and entirely at the back, a bit closer together. This simulation shows some cluster effects after a startup phase clump.

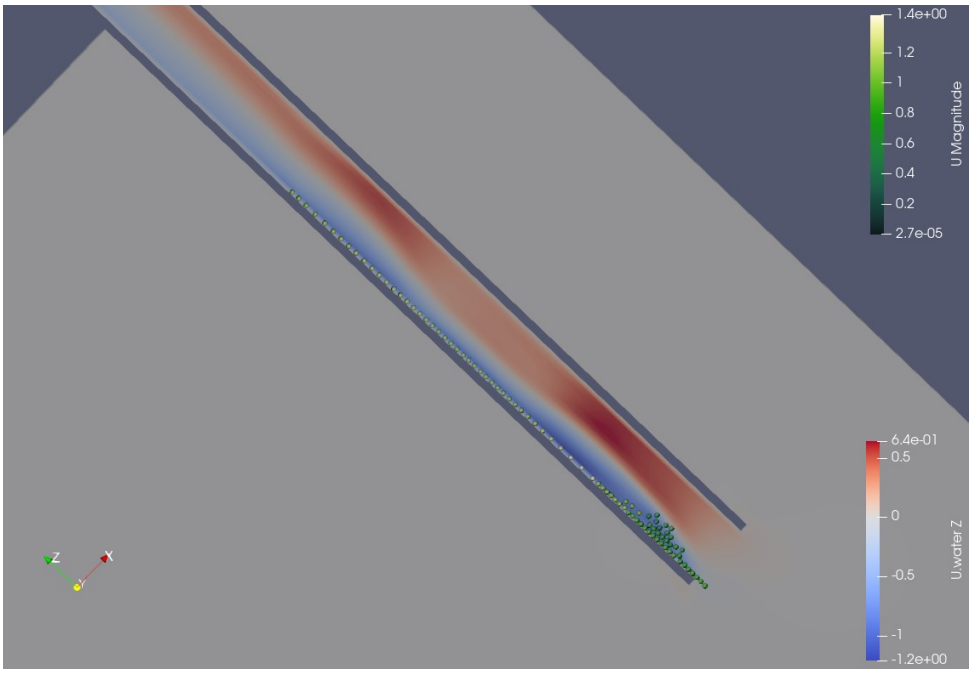


Figure C.18: Rocks in a 2D IFP simulation. With the IFP at an angle of 45 degrees.

In the 60-degree simulation, figure C.19, the rocks in front of the group create a clump with a height of half the pipe diameter. The particles behind the clump are evenly spaced.

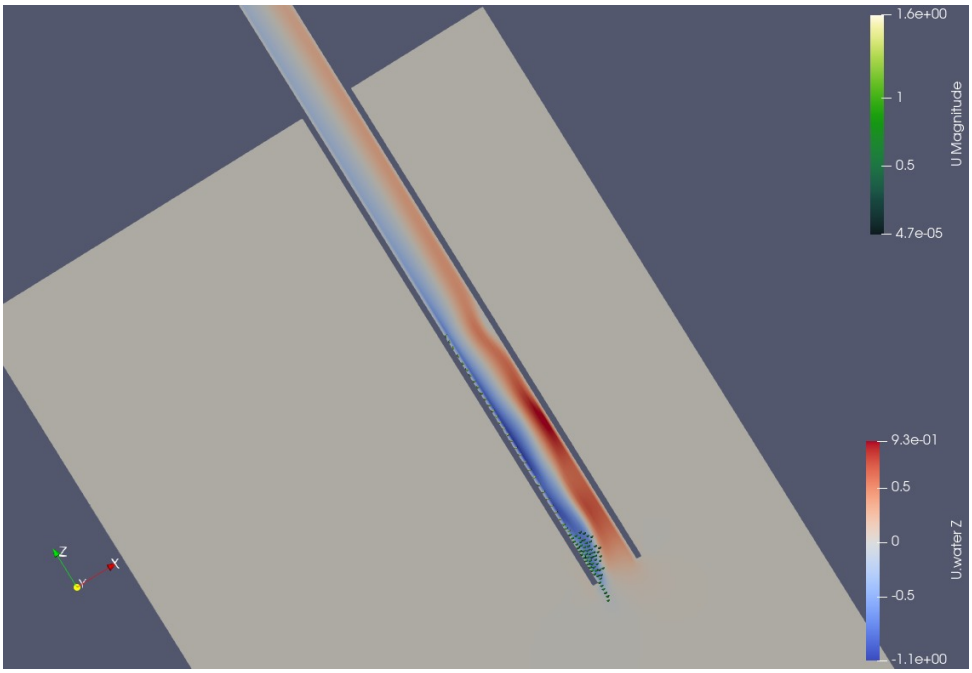


Figure C.19: Rocks in a 2D IFP simulation. With the IFP at an angle of 60 degrees.

In the 75-degree simulation, figure C.20, the rocks in front of the group create a clump larger than half of the pipe. As a result, the particles on the top of this clump move back up the rock stream.

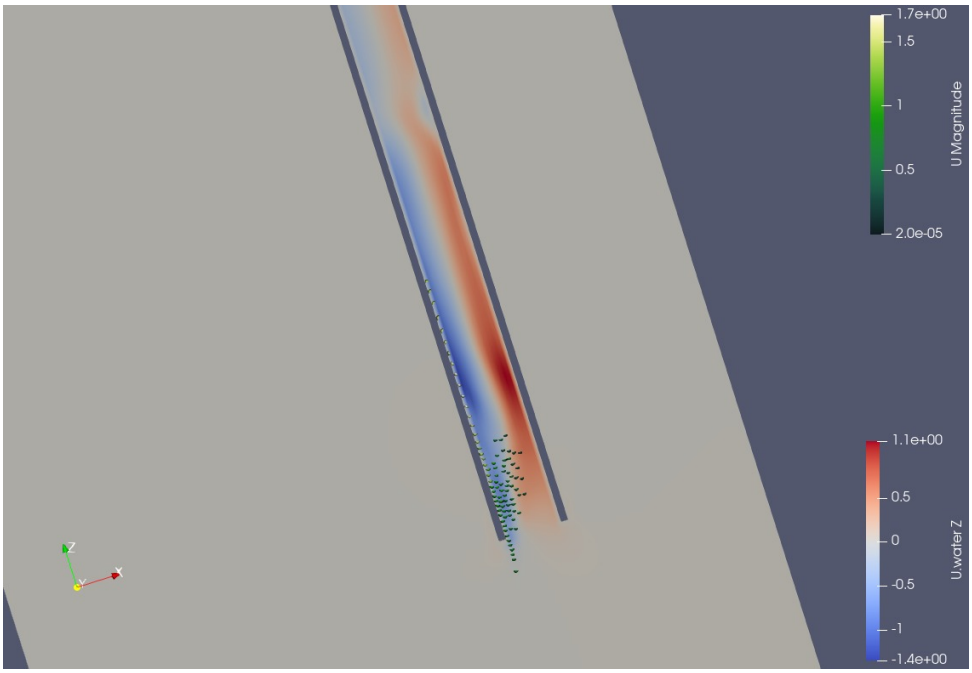


Figure C.20: Rocks in a 2D IFP simulation. With the IFP at an angle of 75 degrees.

The 90-degree simulation, figure C.21, starts with a stream of particles in the middle of the pipe. Before the end, the particles in front are overtaken by the ones behind them and are pushed to the sides. The result of this movement is an outflow over the entire pipe cross-section.

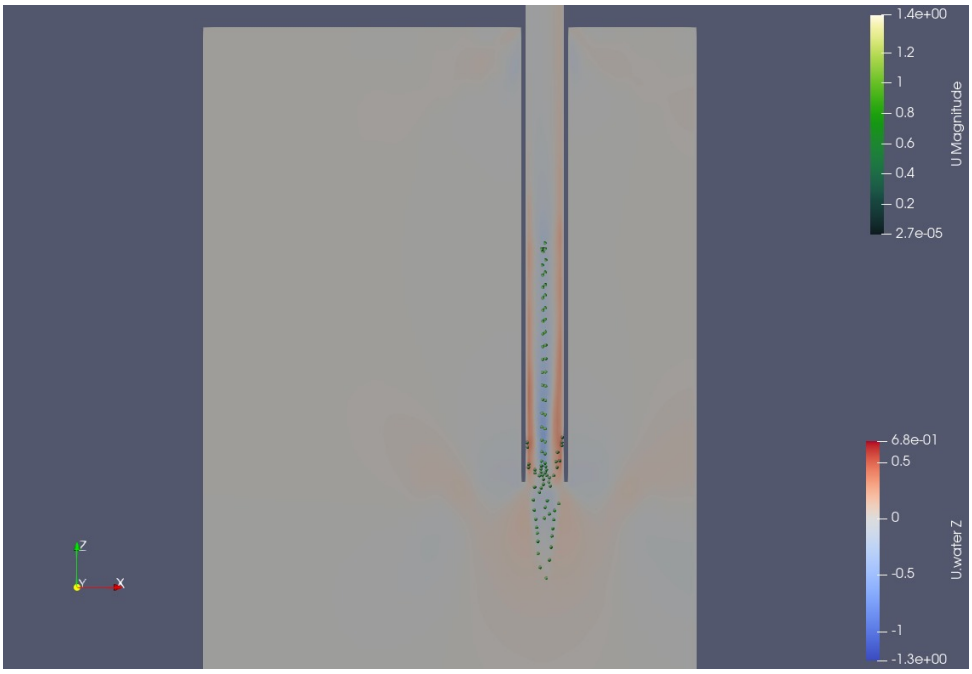


Figure C.21: Rocks in a 2D IFP simulation. With the IFP at an angle of 90 degrees.

The simulations MS30 and MS75 are performed with the same parameters as the 30-degree and 75-degree simulations shown in figure C.16 and C.20, but with more stones added. These extra stones are placed lower in the pipe to create the startup effect, and the stones higher in the pipe do not have to start the flow. The main result of these simulations is that the rocks almost do not show any cluster effect. Reasons could be that the rocks are too few, too evenly spread, and/or too small to experience the influence of the backflow.

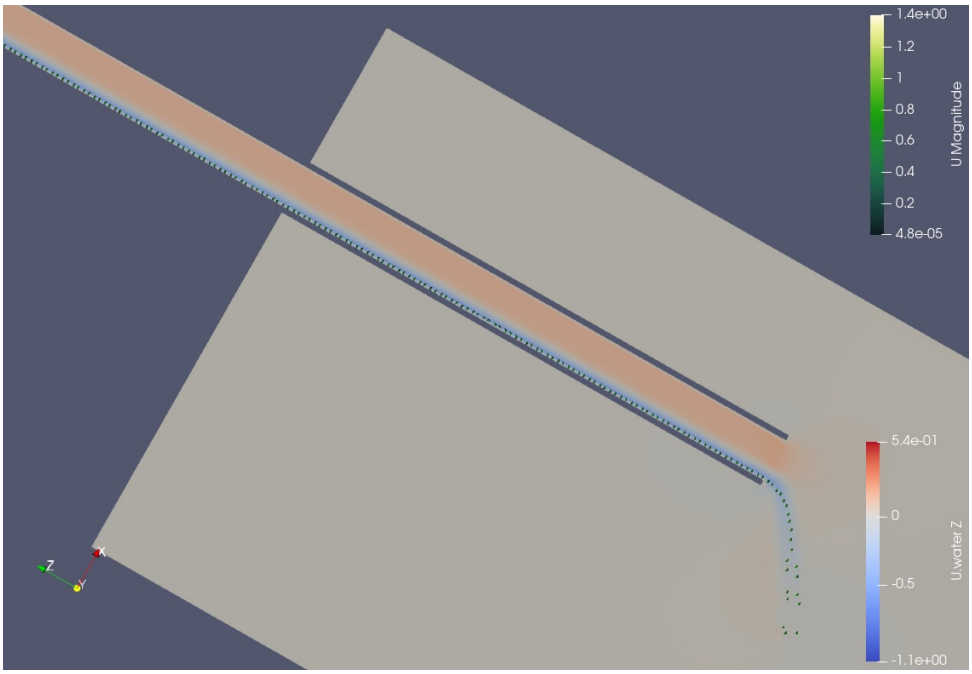


Figure C.22: Rocks in a 2D IFP simulation. With the IFP at an angle of 30 degrees.

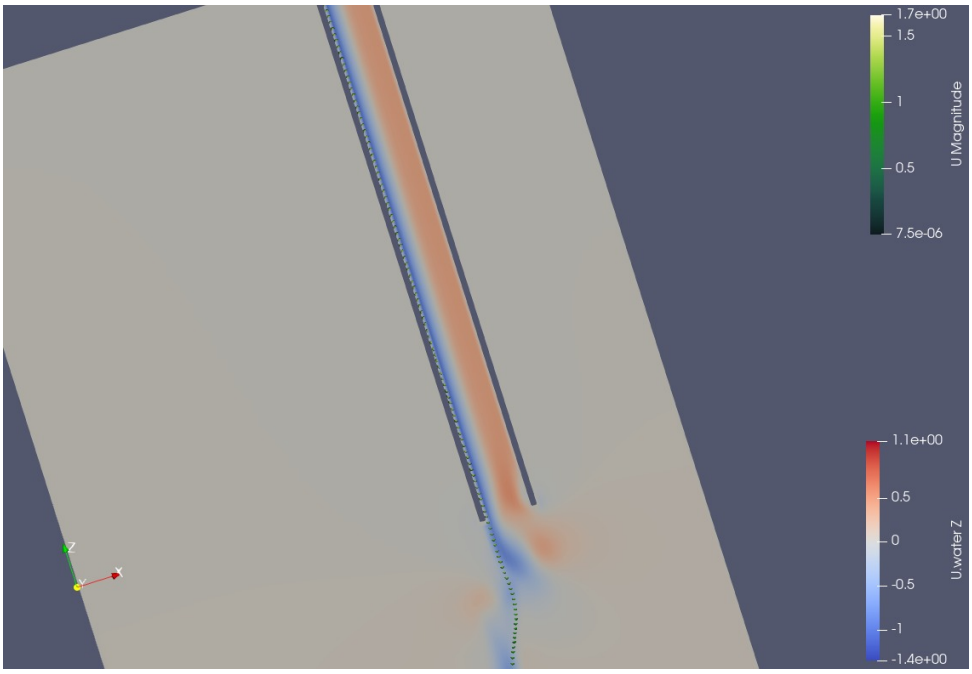


Figure C.23: Rocks in a 2D IFP simulation. With the IFP at an angle of 75 degrees.

Simulations LD75 and RP75 are simulations with a 75-degree pipe. The difference is that the LD has larger rocks. The rocks in this simulation have a diameter of 20 mm. The simulation RP has the rocks randomly placed. The RP simulation also has a larger volume of rocks.
The larger rocks stay in an even-spaced line. No cluster behavior can be seen. A problem with the larger stones simulation is the creation of pulses in the velocity near the edge of the pipe at the exit. One reason for this could be that the area affected by the spread of the particle is larger than the pipe wall thickness. This is a point for further investigation.

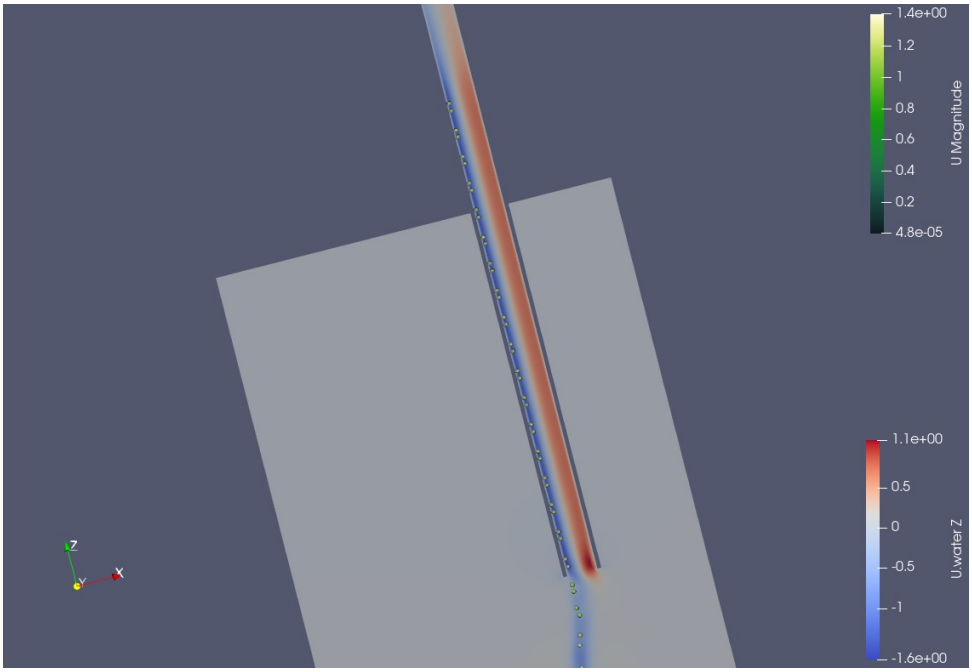


Figure C.24: Rocks in a 2D IFP simulation. With the IFP at an angle of 75 degrees. The rocks are larger than in previous figures.

The simulation with more rocks, which are also irregularly placed, in figure C.25 shows the formation of groups of stones. Between the groups, only some single rocks are present.

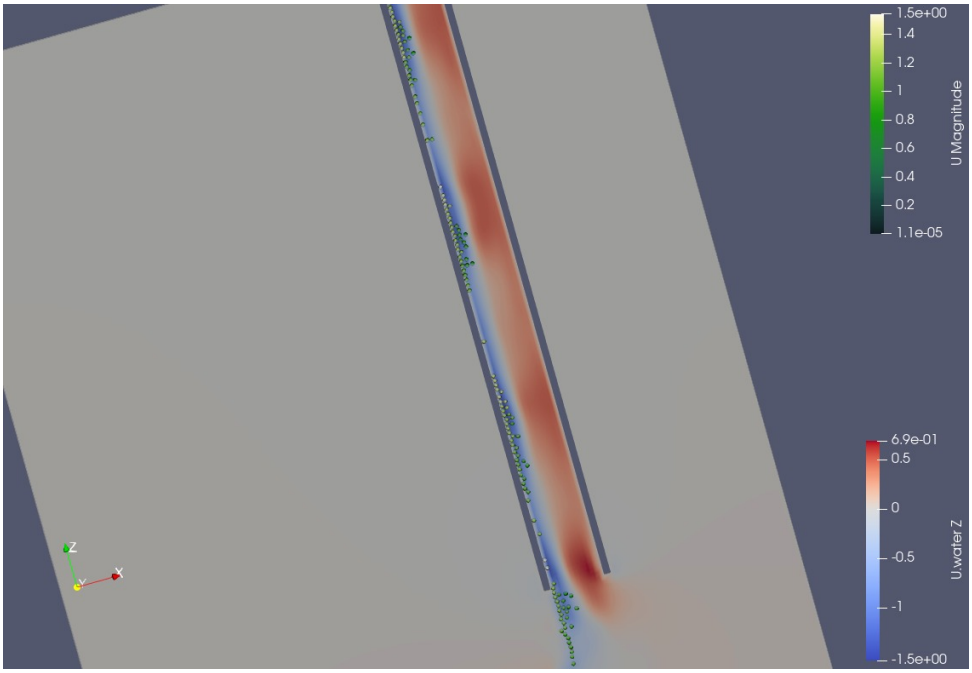


 Figure C.25: Irregularly placed rocks in a 2D IFP simulation. With the IFP at an angle of 75 degrees.

C.2.3 3D simulations

DriftFlux

Simulations in the 3D grid have been performed in gradual succession of difficulty. The first simulation was performed with DriftFluxFoam. The DriftFlux(Foam) solver simulates a density stream by approximating global stone-fluid mixture properties. Individual forces and movements of the stones are not calculated. This simulation tests the grid and walls with a flow comparable to the flows created by the falling particles.
The simulation afterward was a simulation with a single stone. By simulating the single stone, the behavior of the rocks bouncing and rolling on the pipe wall could be investigated.
After one single stone, a simulation with more stones was performed. Since this simulation had a very large calculation time, another simulation with the same number of particles but with a coarser grid in the pipe was performed.

The DriftFlux simulation was performed with the pipe under 75 degrees to horizontal. In figure C.26, the solids concentration is shown after the steady state in the pipe was reached. The concentration of solids enters the pipe entirely at the top. When flowing down, the solids compact along the lower wall up to the packing limit of about 40 percent. After flowing out of the pipe exit, the solid concentration decreases as the solids diverge during the free fall in the water column.

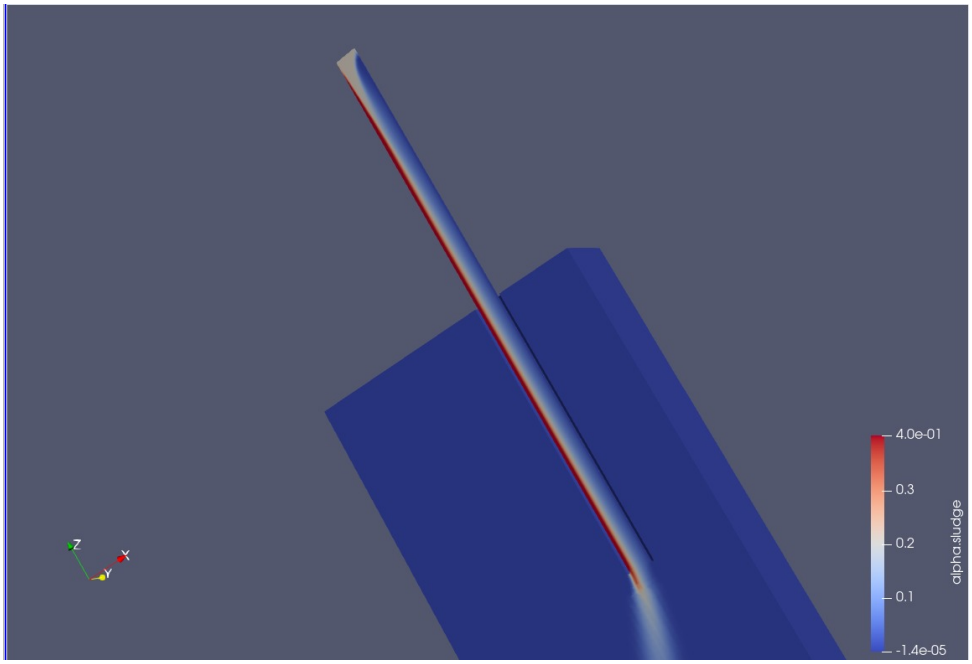


Figure C.26: The solid concentration field for a DriftFlux simulation of an IFP at an angle of 75 degrees.

Figure C.27 shows the velocity field in the z-direction. This is the flow along the pipe. In places with a high concentration, the flow is firmly into a negative z-direction. This is expected as the higher concentrations have a larger density than the surroundings. A small backflow is present along the upper wall but is not larger than the different oscillations in the surrounding waters. A larger backflow would generally be expected. There is no flow through the pipe walls as expected.

The flow differs slightly from the rock simulations due to the inflow of the mixture at the top of the pipe. The simulations with rocks do not have any volume flowing in at this location. This results in an average flow down the pipe instead of the zero volume flux over the pipe cross-section.

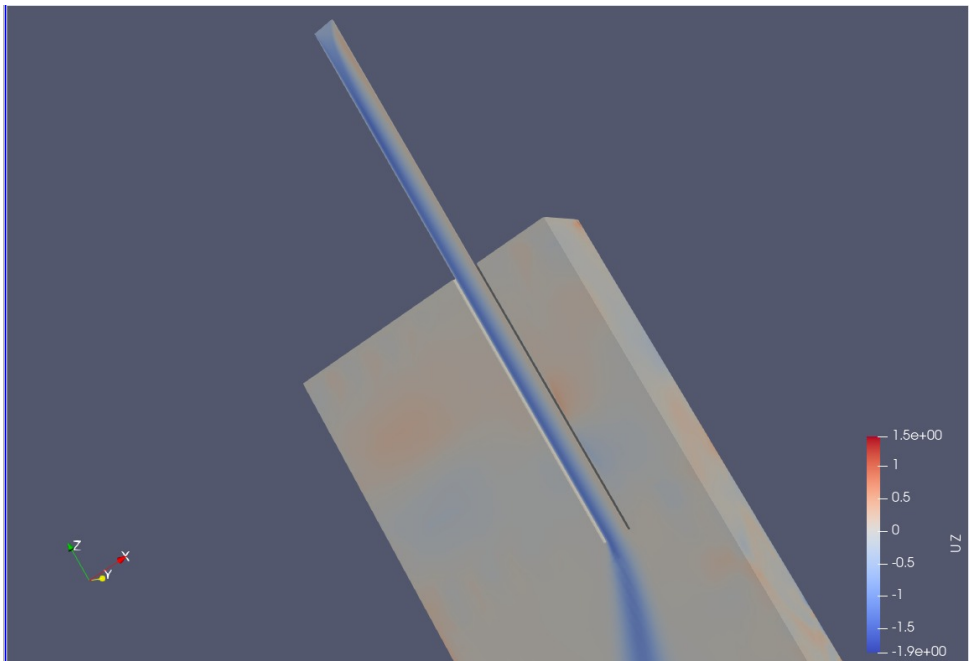


Figure C.27: The flow field in the Z-direction (along the pipe) for a DriftFlux simulation of an IFP at an angle of 75 degrees.

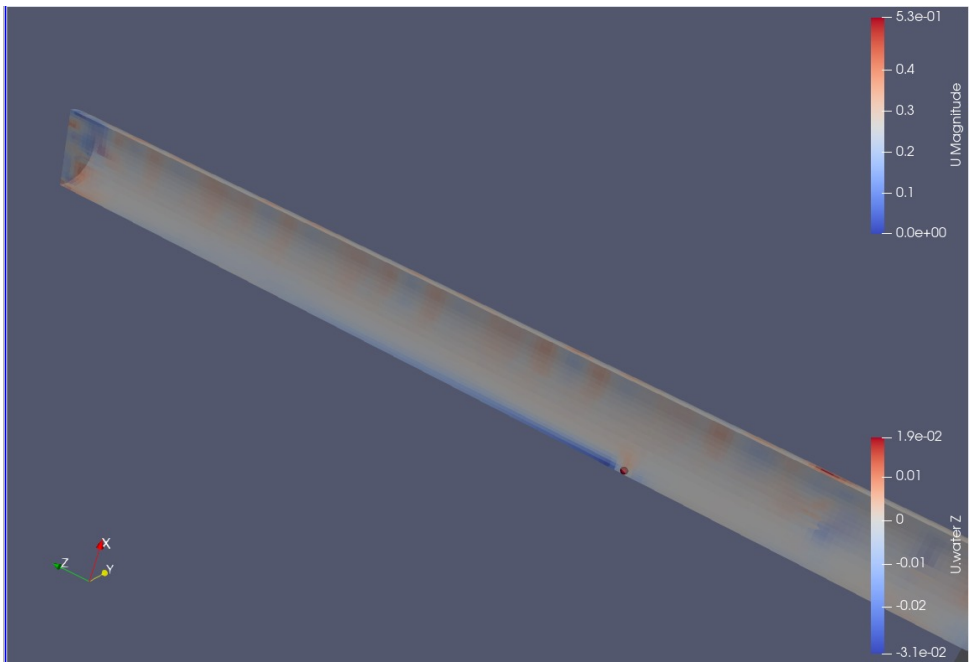


Figure C.28: A simulation of a single particle in a pipe at an angle of 30 degrees. Closeup of the pipe region with the particle inside.

Single particle

Figure C.28 shows the particle and surrounding flow of the single particle simulation. Figure C.29 shows the velocities of the water further away from the particle. Although the flow just around the particle seems plausible, there is a lot of disturbance or turbulence in the pipe's upper layer and around the outside of the pipe. The disturbances outside the pipe are not expected, especially since the particle is still far from the pipe exit.

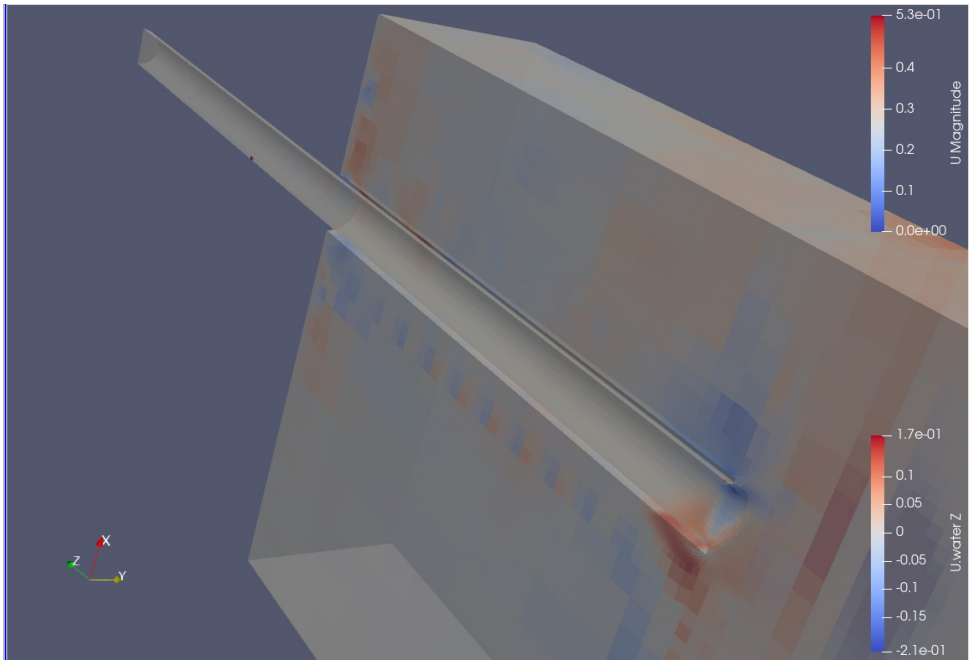


Figure C.29: A simulation of a single particle in a pipe at an angle of 30 degrees. A wider view of the water velocities further away from the particle.

Multiple particles

Figure C.30 shows the results of a 3D simulation of an IPF under 30 degrees with the horizontal with 67 particles. These particles in the front of the group clump together while the particles further back in the group stay in an orderly line, just like in the 2D simulation. Deviations in the flow field around the pipe, as seen with the single particle, are still visible but are less prominent due to the higher flow velocity of the water around the particles. The calculation time for 67 particles in this grid was about 106 hours for 10 seconds of simulated time. This is longer than desired.

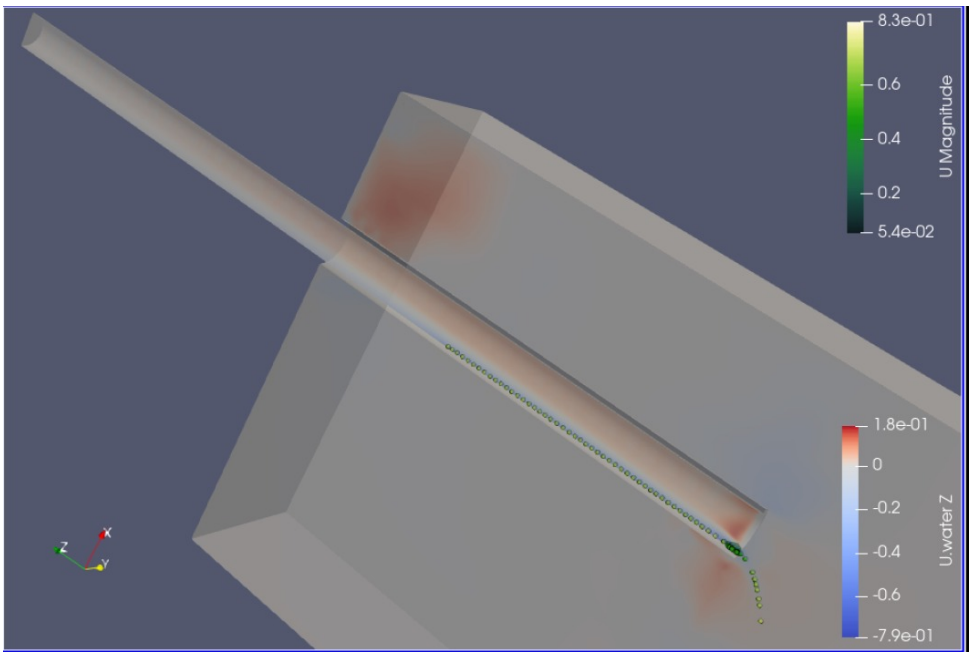


Figure C.30: Results of the 3D simulation with the finer grid. The pipe is under an angle of 30 degrees.

Second, more coarse, grid

Since the simulation times of the first 3D grid were too high, a second 3D grid has been created. The part of the grid in and just around the pipe is one step more coarse in the new grid. 7.5mm in all directions instead of 3.75mm. A simulation with the same parameters as the other 3D simulation was performed. Table C.1 shows that calculation times were reduced significantly.

Results of both simulations with the different grids are compared in figure C.31, figure C.32, and figure C.33. The coarser new grid is on the left in all three figures, and the finer older grid is on the right. The difference in the surrounding water velocities is visible in figure C.31. The velocities outside the pipe especially seem more prominent in the coarser simulation. It is not sure if these velocities are correct or if they are instabilities created due to boundary conditions or large cell-size differences.

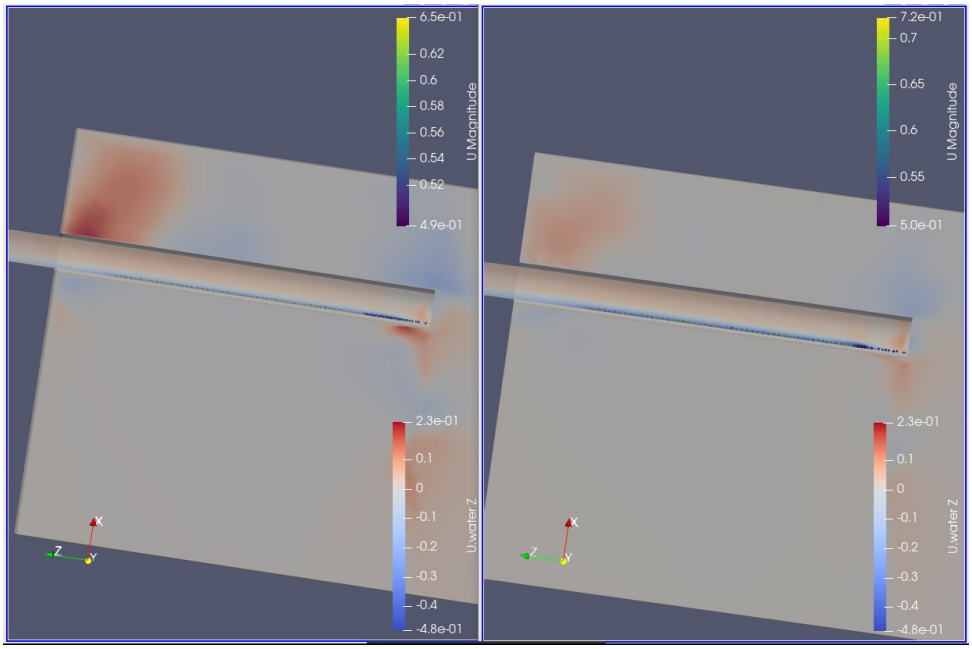


Figure C.31: Comparison between the 3D simulations with a coarser grid (on the left) and the finer grid (on the right). In this view, the flows around the pipe can be compared.

The cluster behavior of the particles of the two simulations can be compared using figure C.32. Both simulations have groups forming in the front of the line of particles. However, the group in the finer grid is more compact as particles are also grouped sideways. In the coarser grid, the particles are all in a back-to-back line. The particles in the more compact group and the surrounding water have a higher velocity than the identical particles and water in the other simulation, which can explain the higher group concentration of these particles.

This higher velocity is also visible in the mean particle velocity comparison in figure C.33. The velocities of the startup phase look the same. The more compact group in front can explain the higher maximum velocity in the middle of the simulation up to when the first particles leave the pipe (the green line in figure C.33). These different groups also influence the particle velocities after the particles exit the pipe, which can explain the differences in the velocities after the green line.

It is not sure which one of the simulations is the better representation. The volume of particles is lower than any test done by Jasper, so it is not possible to compare to his data. However, the coarser grid is the only way to scale up in particle volume. The fine grid will easily take too much time for a simulation with significantly more particles.

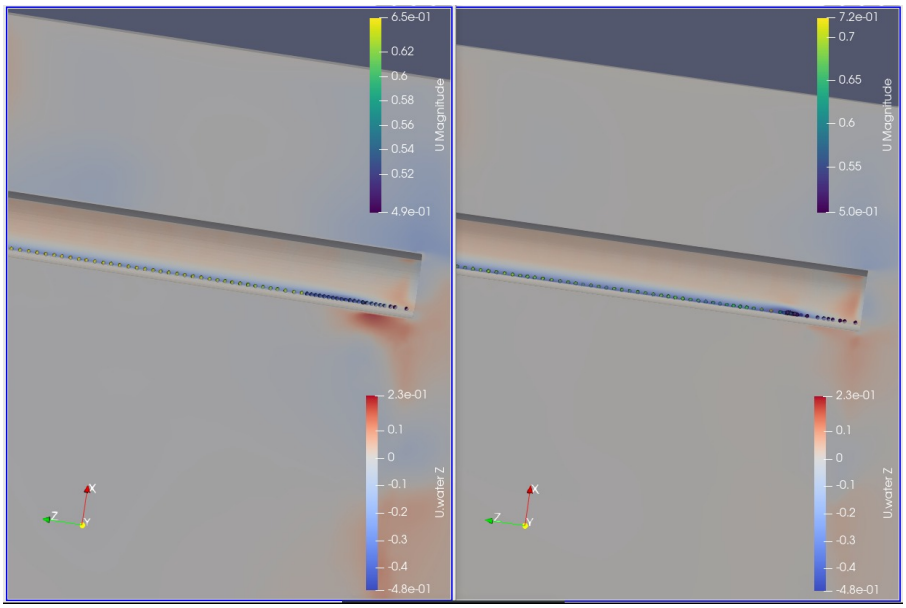


Figure C.32: Comparison between the 3D simulations with a coarser grid (on the left) and the finer grid (on the right). In this view, the particle cluster behavior can be compared.

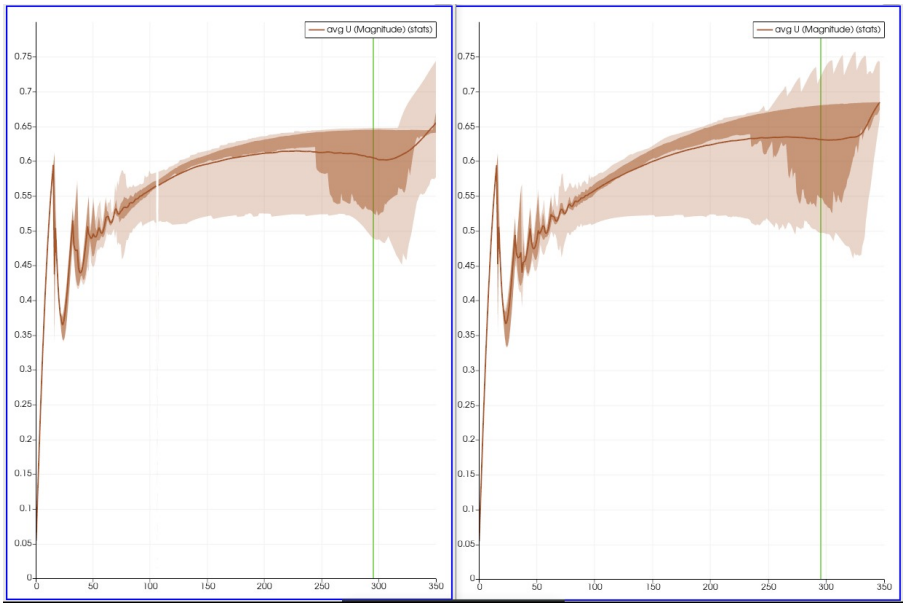


Figure C.33: Comparison between the 3D simulations with a coarser grid (on the left) and the finer grid (on the right).

Domain size studies by flowlines

The flowlines are visualized from the pipe's exit when the first particles exit the pipe. Both for the flow entering as exiting the pipe. These flowlines can be seen in Figure C.34 and Figure C.35. The large straight outflow can be seen especially. The flow into the pipe comes primarily from above the pipe exit. This flow does not reach further than halfway up the immersed part of the pipe. This means the domain up from the exit can be shortened, reducing grid cells and calculation time. The flow seems to be quite turbulent. As well as inside and outside of the pipe, the flow hardly has any straight lines.

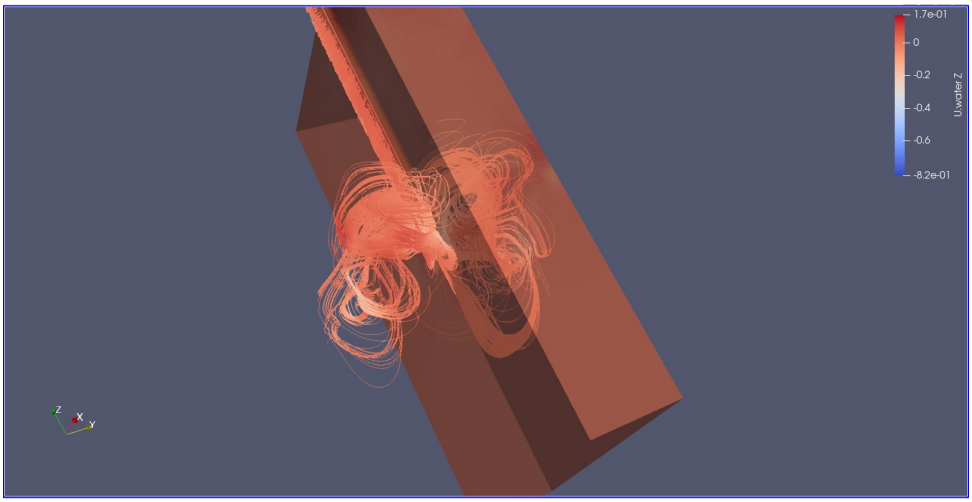


Figure C.34: The flow lines are caused by the out-flow and back-flow of an IFP. The angle of the pipe is 30 degrees.

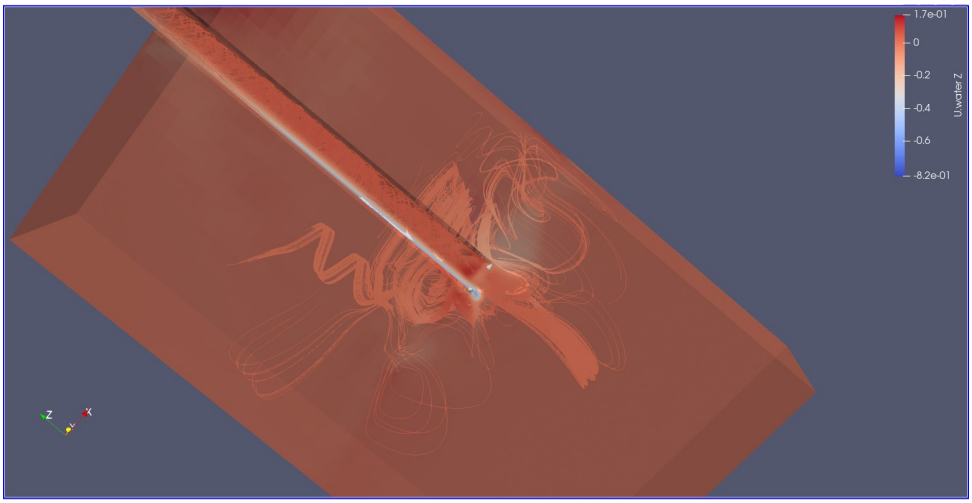


Figure C.35: The flow lines are caused by the out-flow and back-flow of an IFP. The angle of the pipe is 30 degrees.

Third, more uniform, grid

The third grid is more uniform, meaning it contains fewer transitions from smaller to larger grid cells. This should reduce the disturbances in the velocities in and around the pipe. As seen in Figure C.36, the disturbances in the pipe are about the same. This is understandable as the grid in the pipe has hardly changed. Figure C.37 shows the irregularities outside of the pipe. These are about half the velocities of the earlier version of the grid. The more uniform grid does help, but other options, like settings of the relaxation times, might also help. Another option is to find a program that can generate an actual gradient in the cell sizes instead of a jump of 1:2.

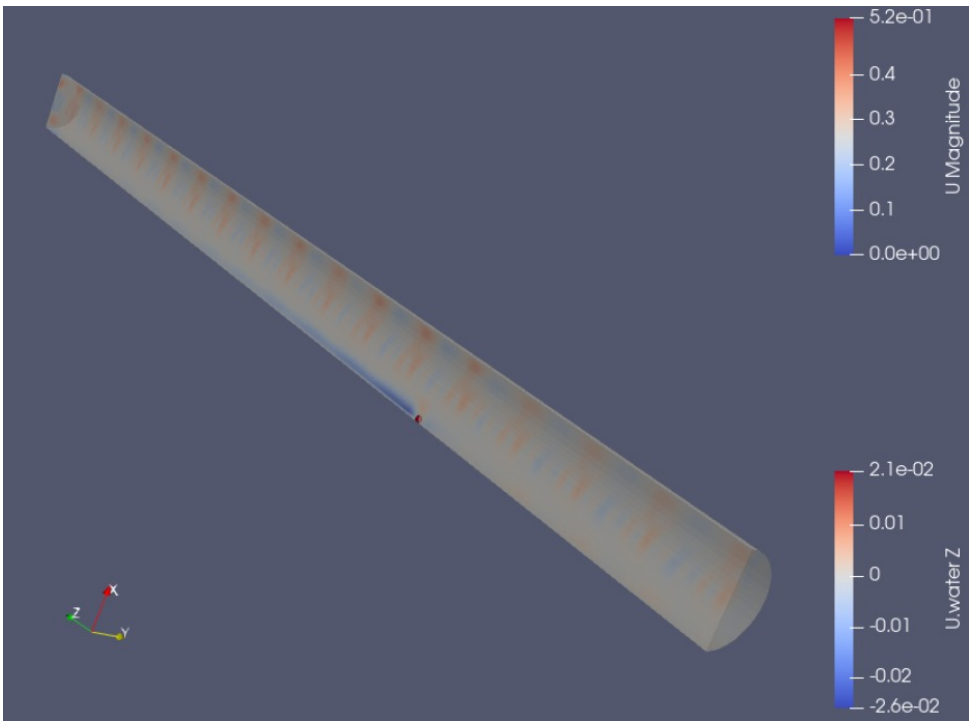


Figure C.36: A simulation of a single stone in the third version of the grid. Only the inside of the pipe is visible.

The simulation with multiple particles, which can be seen in Figure C.38 seems to be as good as the same as with the coarser grid. The particles are again in a line behind each other, with a group close together at the front. The average velocities also follow the same pattern as with the coarser grid.

C.2.4 Simulation results

Single particles

The single particle's 2D and 3D simulation results appear very similar. In Figure C.39, the 3D particle is a bit faster, which can be seen by the fact that it is a bit further down the pipe. In figure C.40, the velocities hardly differ. Further differences in the render can be found in the flowfield. The 3D flowfield has large irregularities. Besides the irregularities, it can be seen that the wake and backflow are both slower in the 3D simulation. This is understandable, as the cross-area ratio of the particle compared to the tube is larger for the 2D pipe than for the 3D pipe.

When looking at the velocities of the particles in 2D and 3D, the largest part is about the same. They both bounce the same in the beginning and have the same strange acceleration of around 170 seconds when they go through the 0-z plane. The only thing that really differs is the behavior near the end of the pipe, around 370 seconds. The particle in the 2D simulation accelerates as it can now fall straight down. The flow field first slows down the particle in the 3D simulation near the end of the pipe. This flow field shouldn't be there and is a result of the grid or the calculation settings.

Comparing both velocities to the velocities measured by Fontijn in the model tests in Figure **??**, will show that the velocities from the simulations are overestimations of the particle velocity. The most significant

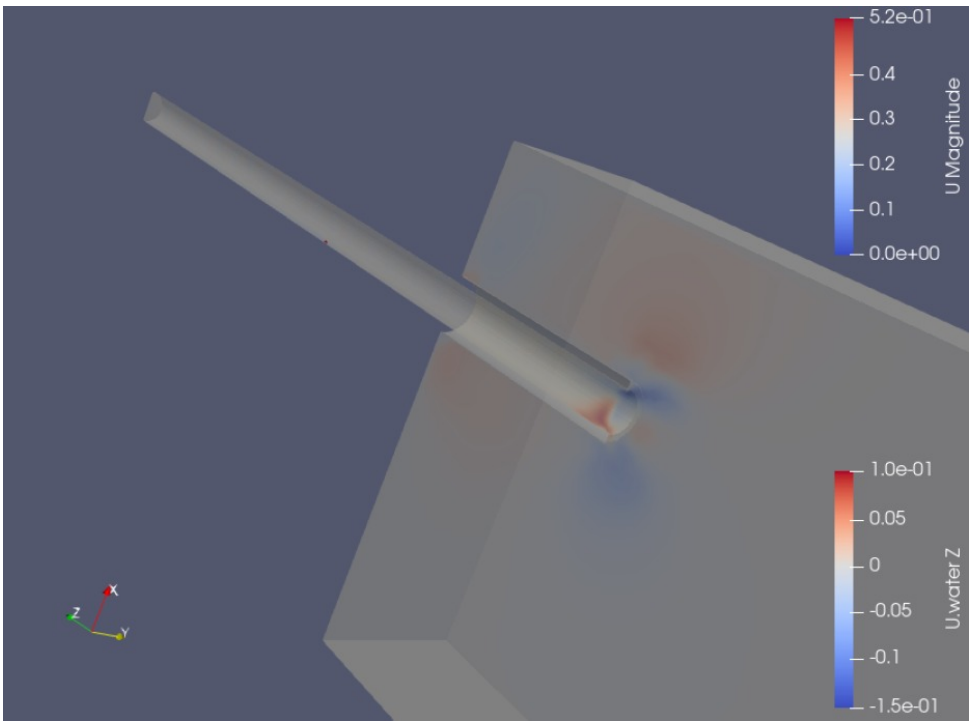


Figure C.37: A simulation of a single stone in the third version of the grid. Zoomed out for a better view of the flow in the region outside of the pipe.

contributor to this overestimation is that the rotational velocity is not used in the drag calculations. Due to the rotational velocity, the upper side of the particle has a larger downward velocity than the average particle velocity. Thus, it has a larger velocity compared to the fluid around it. Not taking this into account will result in underestimating the drag force. With spinning free-falling particles, the difference in drag is compensated by the other side of the particle, which moves slower than the particle. However, as the particle is rolling, this part interacts with the wall and less with the fluid.

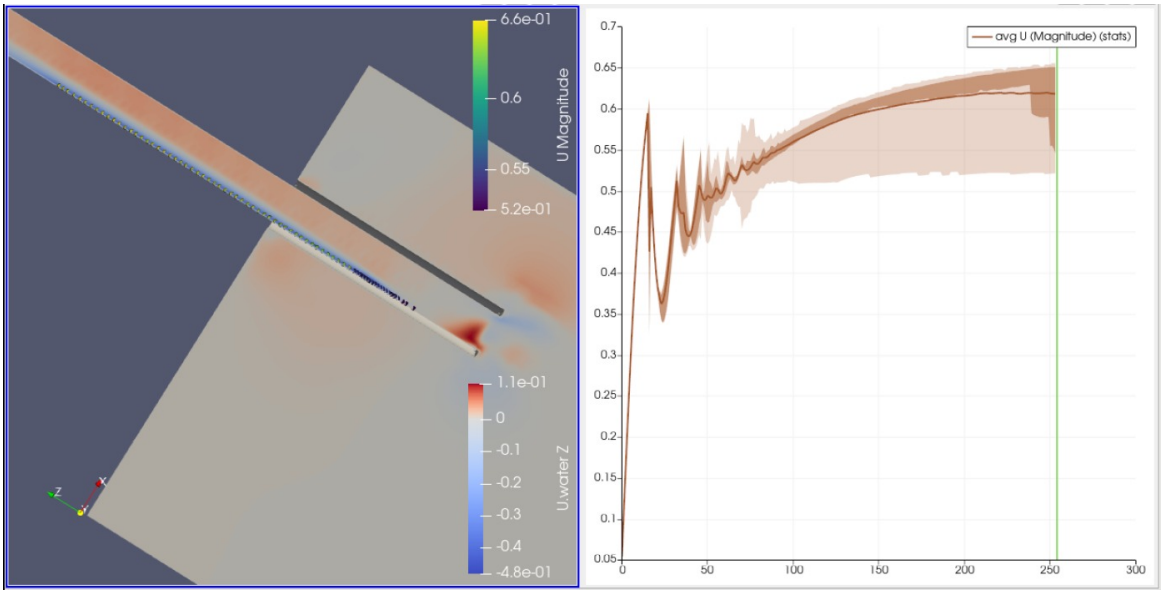


Figure C.38: A simulation of 89 particles in a 30-degree IFP with the third version of the grid. On the right is the average velocity of the particles plotted over time.

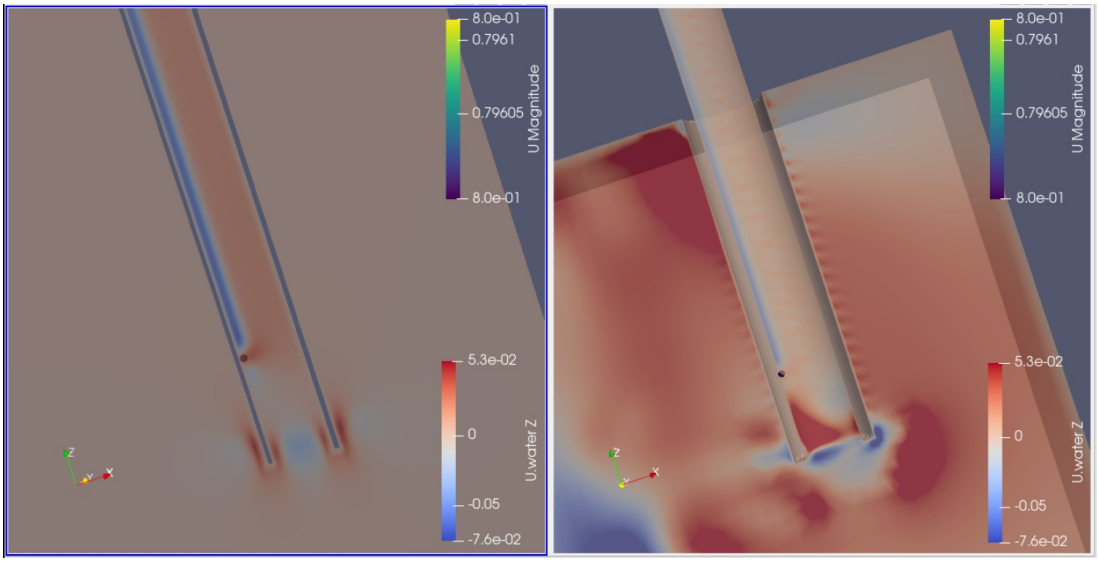


Figure C.39: A simulation of a single particle in a 75-degree IFP. On the left in 2D, and on the right in 3D. Both are on 3.5 seconds of the simulation.

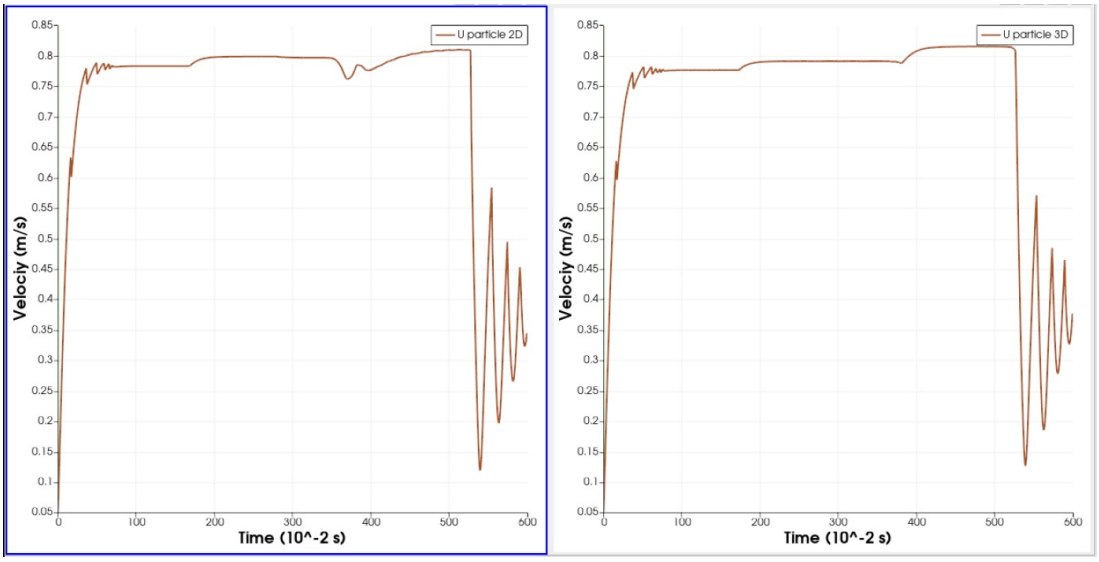


Figure C.40: The velocities of a simulated single particle in a 75-degree IFP. On the left in 2D, and on the right in 3D.

The acceleration around z=0

In figure C.40 an acceleration of the particle could be seen halfway the pipe. This was around z=0 plane in both the 2D as 3D simulation. In order to check if this was due to the negative z value or something else, another 2D simulation was performed. The new simulation had a grid which was moved 0.55 meter in negative z direction. This way the particle would move through z=0 halfway between the bouncing part and the acceleration of the old simulation. See Figure C.41 for the exact location.

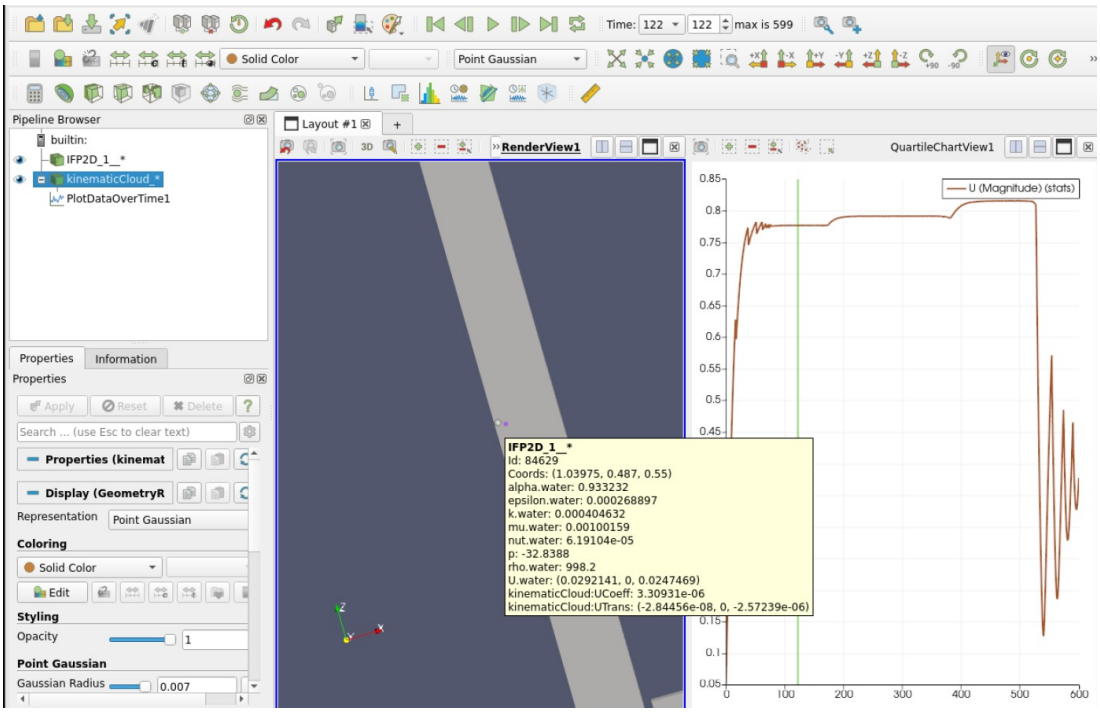


Figure C.41: In this figure the movement in negative z-direction is determined.

The velocity comparison between the old simulation and the simulation with everything moved into negative z-direction can be seen in Figure C.42. It shows that the acceleration is not caused by the negative z-value.

When looking at the torque C.43, it can be seen that the moment of acceleration corresponds with the moment the torque decreases. The decreasing torque can be explained using Figure C.44, from the time the torque decreases, the particle is rolling exactly fast enough that it does not slip anymore. The decreasing resistance force of the wall on the particle explains the acceleration of the particle.

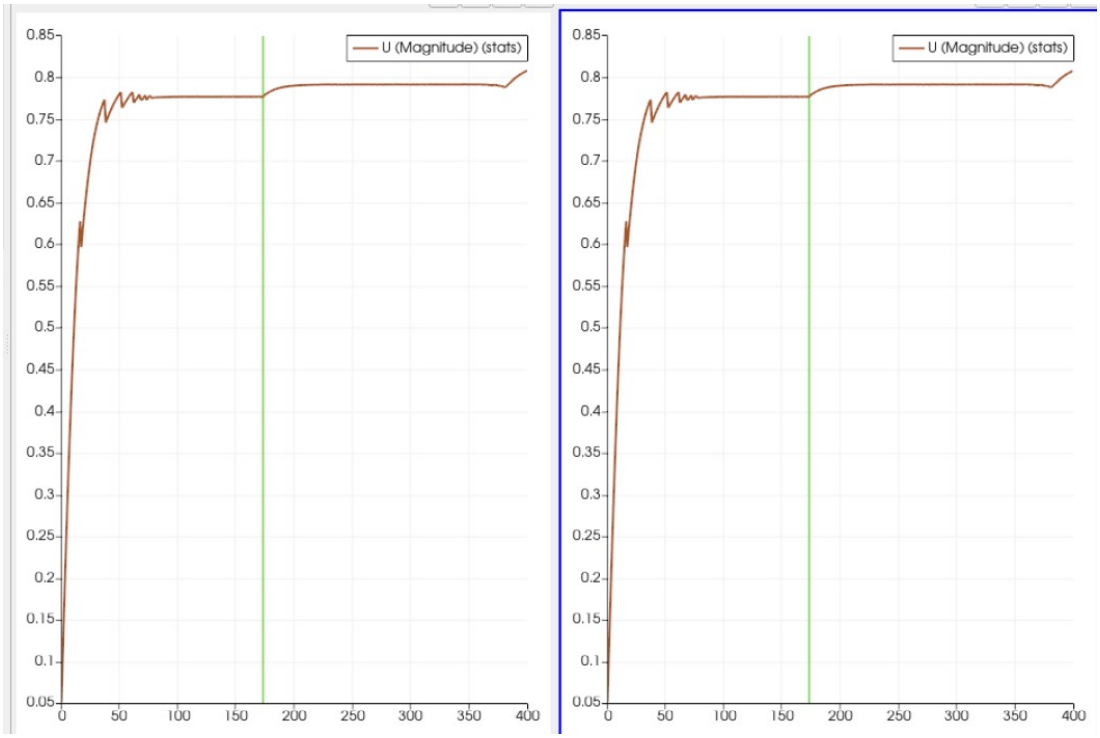


Figure C.42: A comparison of the velocities of a single particle in the original simulation (on the left) and a simulation of which everything was moved 0.55 meter in the negative z-direction (on the right).

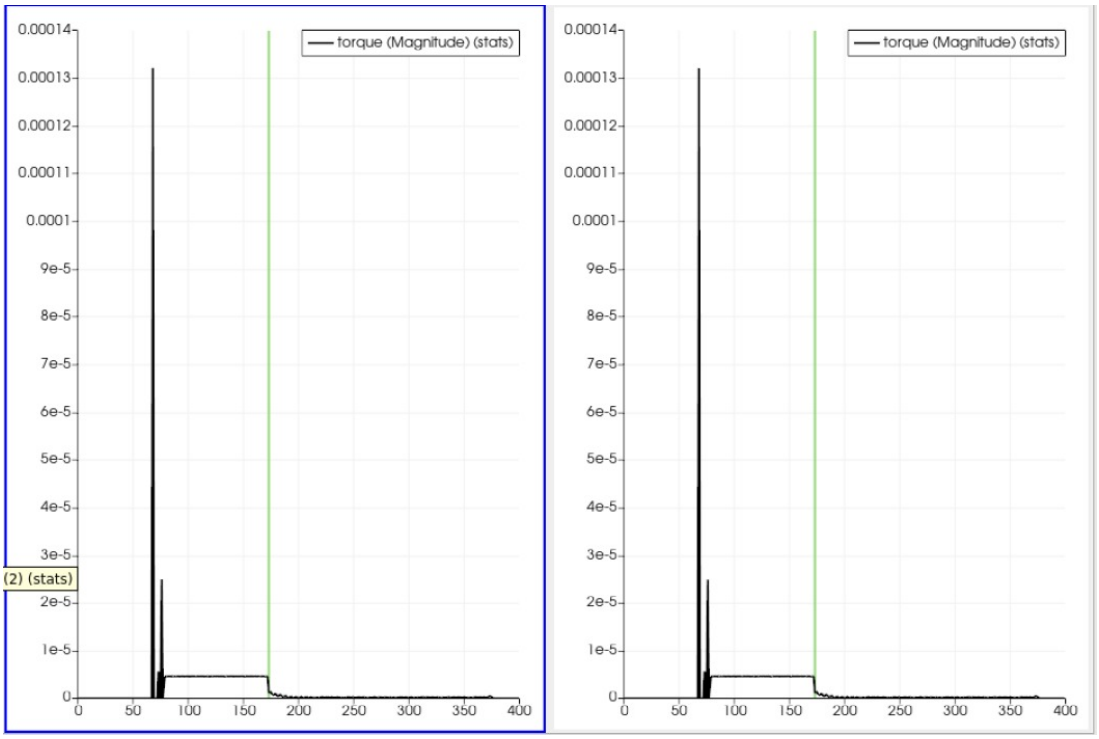


Figure C.43: The torque experienced by the particle in the simulation in the original simulation (on the left) and a simulation in which everything was moved 0.55 meters in the negative z-direction (on the right).

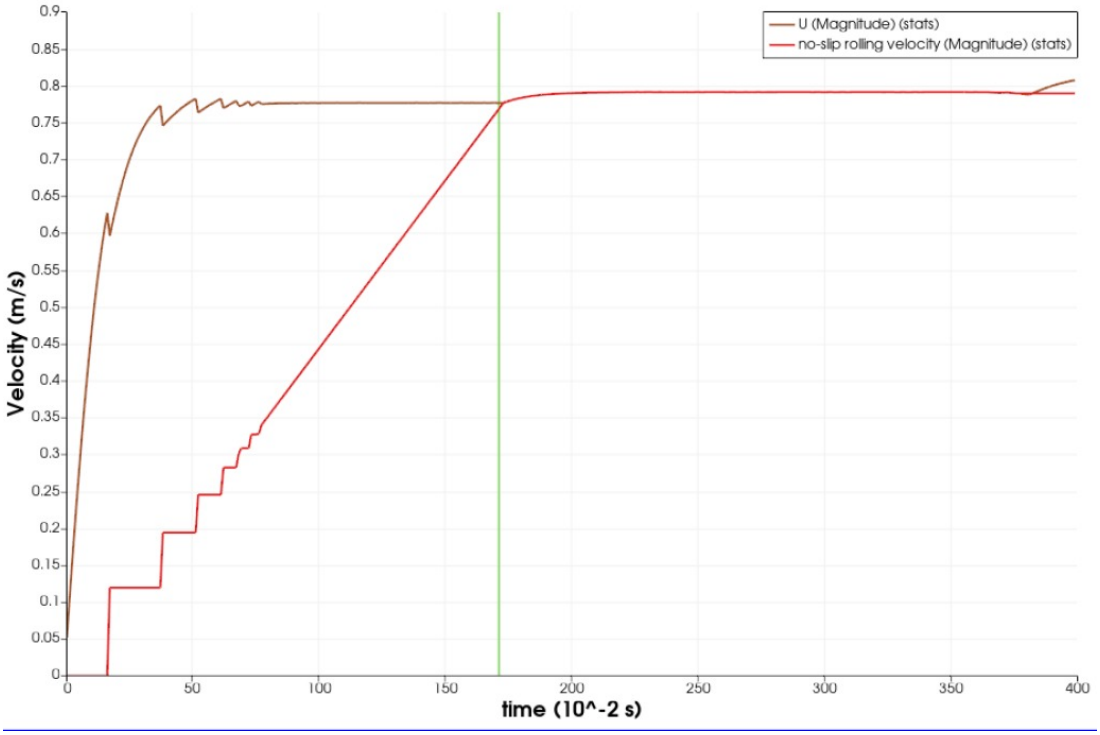


Figure C.44: The linear velocity of the particle in case it would not slip compared to the velocity of the particle.

D Finding the value for C_{factor}

The tests for single stones consist of 3 steps.
The first step is finding the C_{factor} for the particle not falling in a free water column but instead rolling along the pipe wall.
The second step is finding the C_{factor} value for a non-spherical particle moving along the pipe wall
The third step is to run the simulations with the earlier found C_{factor} value for the particle sizes between 8 and 34 mm.

D.0.1 C_{factor} for rolling along a wall

Particles rolling along an inclined pipe wall experience a higher C_d value than when falling through a free water column, as discussed in section 3.2.5. Different values for the C_{factor} have been used in simulations to find the velocity matching the marble from Fontijn's lab tests. The parameters used for the study can be found in table D.1. These simulations are performed in 2D.

Table D.1: Parameters of the C_{factor} variation simulations for the wall effect.

	Value	Unit
Angle	75	degrees
ρ_p	2500	kg/m^3
d_p	14	mm
C_{factor}	1, 1.5, 1.6, 1.8, 2	-

The equilibrium rolling velocity results are in table D.2 and plotted against the marble velocity of 0.64 m/s in figure D.1.

As can be seen in the figure D.1, the C_d of the marble rolling along the pipe wall is about 1.6 times higher than the C_d for a particle in the free-water column. A C_d of 10 times higher was reported by Chhabra, Kumar, and Prasad [12] for particles with a $Re_p < 1$ and a C_d 3 times higher for $250 \leq Re_p \leq 3000$. The particle with the C_{factor} of 1.6 has a Re_p of about 8900, which lies above the Reynolds range for which Chhabra, Kumar, and Prasad performed their tests. As for higher Re_p values, the difference in C_d value seems to drop, and with the found C_{factor} between 1 for a fully free-falling particle and 3 for a particle rolling with a lower particle Reynolds number, the found value is not difficult to believe. However, a factor of 3 in Reynolds number is not a large difference, so a value closer to 3 was expected.

D.0.2 C_{factor} variations for rocks

Rocks experience a larger drag force while settling than spheres of the same volume. This is due to a higher C_d value, as described in section 3.4.1. Different values for the C_{factor} have been used in simulations to find the velocity of a single rock that matches the trend from Fontijn's lab results. The parameters of this study can be found in D.3. All simulations were performed in 2D.

Table D.2: Resulting equilibrium rolling velocities per C_{factor}.

C_{factor}	Velocity (m/s)
1	0.791
1.5	0.655
1.6	0.636
1.8	0.602
2	0.574

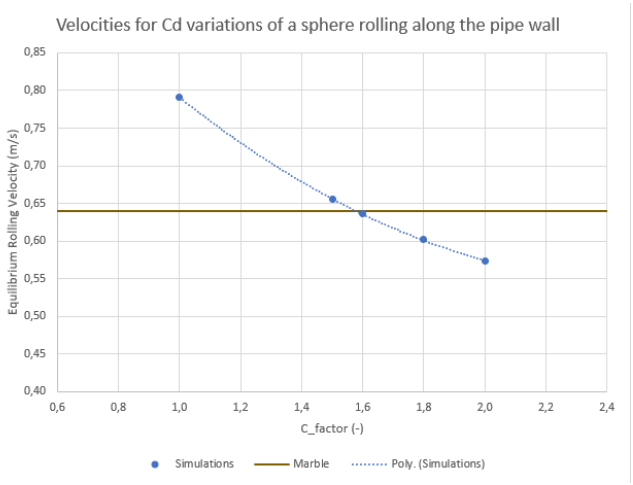


Figure D.1: Results of varying C_{factor} compared to the marble velocity.

Table D.3: Parameters of the C_{factor} variation simulations for rocks.

	Value	Unit
Angle	75	degrees
ρ_p	2670	kg/m^3
d_p	14	mm
C_{factor}	1,2,3,4,5	-

The equilibrium sliding velocity results are in table D.4 and plotted against the rock velocity of 0.48 m/s according to a linear trend-line at 14 mm in figure D.2.

Table D.4: Resulting equilibrium sliding velocities per C_{factor}.

C_{factor}	Velocity (m/s)
1	0.817
2	0.593
3	0.493
4	0.434
5	0.395

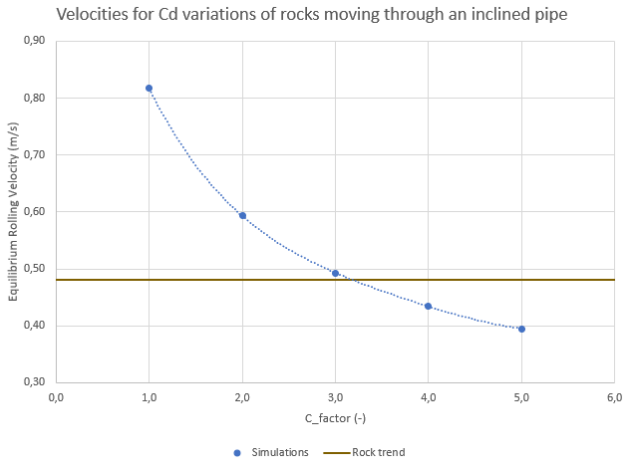


Figure D.2: Results of varying C_{factor} compared to the rock's velocity.

The results show that a total C_{factor} of around 3 gives a velocity that matches the velocity for a rock with a spherical equivalent diameter of 14 mm according to the trend line through Fontijn's data. It is important to realize that the value of 3 includes a factor of 1.6 for the wall the particles move along. The factor remaining

for the compensation of the rock shapes is 1.9. This is way lower than values discussed in section 3.4.1 found by Van der Wall [15], 3 times independent of the shape and 2 times when considering the lowest shape factor. When using the equation of Wu and Wang [19], the value of 1.9 would match particles with a shape factor of 0.8.

D.0.3 Particle diameter variations for the C_{factor} of 3

For the C_{factor} of 3, the velocity of a 14 mm particle matches the rock velocity trend line. By changing the particle size, the limits for which this C_{factor} can be used is investigated. First, the results for 2D simulations with these different particle sizes are shown. All simulations have been done at an angle of 75 degrees. The densities are chosen to match those of the rocks closest to the particle diameter. Table D.5 shows the particle diameters, densities, and resulting equilibrium sliding velocities for these simulations. The velocities are compared to Fontijn's single stone tests in figure D.3.

Table D.5: Velocities of 2D simulations for different size single particles in an IFP under 75 degrees.

$d_p(mm)$	$\rho_p(kg/m^3)$	Velocity (m/s)
8	2711	
14	2670	
20	2670	
24	2900	
27	2900	
30	2900	
34	2900	

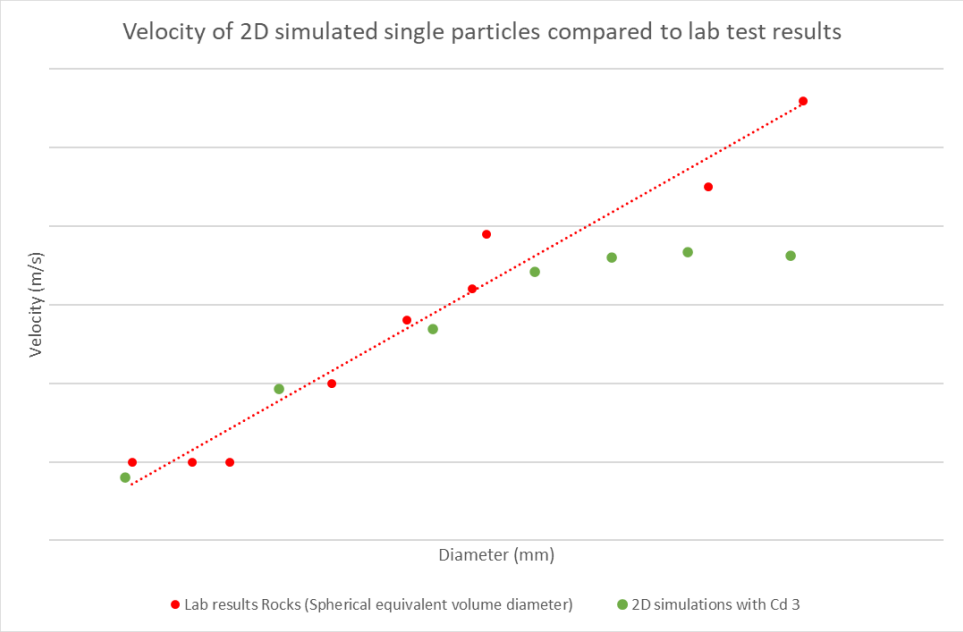


Figure D.3: Velocities of 2D simulations for different size single particles compared to lab tests rock velocities.

The velocities for the particles with a diameter smaller than 25 mm seem to match the velocities of Fontijn's single stone tests. The particles larger than 25 mm are slower than the model lab test data. Part of the reason the particles are going slower might be the wall effect. This effect is stronger in the 2D simulations due to the incorrect cross-sectional area ratio. The cross-section pipe is smaller compared to that of the particle.

Due to the stronger wall effect in the 2D simulations, the same simulations were redone in 3D. Table D.6 shows the particle diameters, densities, and resulting equilibrium sliding velocities for these simulations. The velocities are compared to Fontijn's single stone tests in figure D.3.

The velocities of the particles with a diameter of up to 25 mm do not show any changes. The particles larger than 25 mm are slower than the model lab data but did improve compared to the 2D simulation data. The 27 mm particle velocity lies closer to the trend line than some model test data. The same can not be said for the

Table D.6: Velocities of 3D simulations for different size single particles in an IFP under 75 degrees.

$d_p(mm)$	$\rho_p(kg/m^3)$	Velocity (m/s)
8	2711	
14	2670	
20	2670	
27	2900	
34	2900	

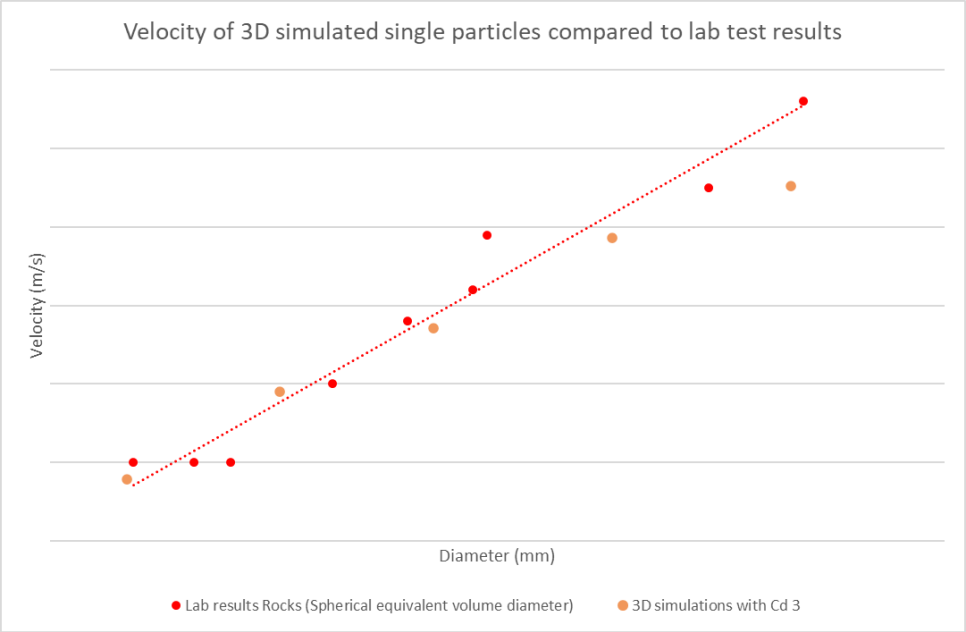


Figure D.4: Velocities of 3D simulations for different size single particles compared to lab tests rock velocities.

34 mm particle. The C_{factor} of 3 seems to fit for particle diameters up to 27 mm. The C_{factor} of 3 appears to be an over-adjustment for particles with larger diameters.

Bibliography

[1] J. F. Richardson and W. N. Zaki. "The sedimentation of a suspension of uniform spheres under conditions of viscous flow". In: *Chemical Engineering Science* 3.2 (Apr. 1, 1954), pp. 65–73. ISSN: 0009-2509. DOI: 10.1016/0009-2509(54)85015-9. URL: https://www.sciencedirect.com/science/article/pii/0009250954850159 (visited on 03/04/2024).

[2] S. I. Rubinow and Joseph B. Keller. "The transverse force on a spinning sphere moving in a viscous fluid". In: *Journal of Fluid Mechanics* 11.3 (Nov. 1961), pp. 447–459. ISSN: 1469-7645, 0022-1120. DOI: 10.1017/S0022112061000640. URL: https://www.cambridge.org/core/journals/journal-of-fluid-mechanics/article/transverse-force-on-a-spinning-sphere-moving-in-a-viscous-fluid/4AC1970527C8C9EAD7D65A412431A492 (visited on 04/25/2024).

[3] P. G. Saffman. "The lift on a small sphere in a slow shear flow". In: *Journal of Fluid Mechanics* 22.2 (June 1965). Number: 2, pp. 385–400. ISSN: 0022-1120, 1469-7645. DOI: 10.1017/S0022112065000824. URL: https://www.cambridge.org/core/product/identifier/S0022112065000824/type/journal_article (visited on 01/22/2024).

[4] Martin R. Maxey and James J. Riley. "Equation of motion for a small rigid sphere in a nonuniform flow". In: *The Physics of Fluids* 26.4 (Apr. 1, 1983). Number: 4, pp. 883–889. ISSN: 0031-9171. DOI: 10.1063/1.864230. URL: https://pubs.aip.org/pfl/article/26/4/883/816404/Equation-of-motion-for-a-small-rigid-sphere-in-a (visited on 01/22/2024).

[5] Robert Iskanderovich Nigmatulin. *Dynamics Of Multiphase Media*. Google-Books-ID: avfwXWT2fqAC. CRC Press, Sept. 1, 1990. 402 pp. ISBN: 978-0-89116-328-2.

[6] H. Sakamoto and H. Haniu. "A Study on Vortex Shedding From Spheres in a Uniform Flow". In: *Journal of Fluids Engineering* 112.4 (Dec. 1, 1990). Number: 4 Publisher: American Society of Mechanical Engineers Digital Collection, pp. 386–392. ISSN: 0098-2202. DOI: 10.1115/1.2909415. URL: https://dx.doi.org/10.1115/1.2909415 (visited on 02/09/2024).

[7] R. Di Felice. "The voidage function for fluid-particle interaction systems". In: *International Journal of Multiphase Flow* 20.1 (Feb. 1, 1994), pp. 153–159. ISSN: 0301-9322. DOI: 10.1016/0301-9322(94)90011-6. URL: https://www.sciencedirect.com/science/article/pii/0301932294900116 (visited on 03/04/2024).

[8] H. J. Herrmann et al. "Density waves in granular flow". In: *Non-Linearity and Breakdown in Soft Condensed Matter*. Ed. by Kamal K. Bardhan, Bikas K. Chakrabarti, and Alex Hansen. Lecture Notes in Physics. Berlin, Heidelberg: Springer, 1994, pp. 28–39. ISBN: 978-3-540-49037-1. DOI: 10.1007/3-540-58652-0_27.

[9] H. J. Cregten. "Het gebruik van probabilistische methoden bij het beoordelen van het stortresultaat van een zijstorter". In: (1995). Publisher: TU Delft, Civil Engineering and Geosciences, Hydraulic Engineering. URL: https://repository.tudelft.nl/islandora/object/uuid%3Aa78ee7da-fae7-424b-99c5-f55f83068ad8 (visited on 02/07/2023).

[10] T. Raafat, J. P. Hulin, and H. J. Herrmann. "Density waves in dry granular media falling through a vertical pipe". In: *Physical Review E* 53.5 (May 1, 1996). Number: 5, pp. 4345–4350. ISSN: 1063-651X, 1095-3787. DOI: 10.1103/PhysRevE.53.4345. URL: https://link.aps.org/doi/10.1103/PhysRevE.53.4345 (visited on 03/02/2023).

[11] J. C. Winterwerp. "On the dynamics of high-concentrated mud suspensions". In: (1999). URL: https://repository.tudelft.nl/islandora/object/uuid%3A0b503064-91ad-48de-8174-761c315f8132 (visited on 03/04/2024).

[12] R. P Chhabra, Mukesh Kumar, and Rishitosh Prasad. "Drag on spheres in rolling motion in inclined smooth tubes filled with incompressible liquids". In: *Powder Technology* 113.1 (Nov. 20, 2000), pp. 114–118. ISSN: 0032-5910. DOI: 10.1016/S0032-5910(00)00211-4. URL: https://www.sciencedirect.com/science/article/pii/S0032591000002114 (visited on 03/03/2024).

[13] G. G. Joseph et al. "Particle–wall collisions in a viscous fluid". In: *Journal of Fluid Mechanics* 433 (Apr. 2001), pp. 329–346. ISSN: 1469-7645, 0022-1120. DOI: 10.1017/S0022112001003470. URL: https://www.cambridge.org/core/journals/journal-of-fluid-mechanics/article/particlewall-collisions-in-a-viscous-fluid/AA9C176CEDAC4CB1D40847D2E68C3F65 (visited on 03/04/2024).

[14] P. Gondret, M. Lance, and L. Petit. "Bouncing motion of spherical particles in fluids". In: *Physics of Fluids* 14.2 (Feb. 1, 2002), pp. 643–652. ISSN: 1070-6631, 1089-7666. DOI: 10.1063/1.1427920. URL: https://pubs.aip.org/pof/article/14/2/643/451572/Bouncing-motion-of-spherical-particles-in-fluids (visited on 03/04/2024).

[15] J. R. Van der Wal. "Het valgedrag van steengroepen: Modelonderzoek ter vergroting van het inzicht in het valgedrag van breuksteen". In: (2002). Publisher: TU Delft, Faculty of Civil Engineering and Geosciences, Hydraulic Engineering. URL: https://repository.tudelft.nl/islandora/object/uuid%3Add449574-1e12-4b2c-a28c-3681b502098d (visited on 04/16/2024).

[16] Phillip P. Brown and Desmond F. Lawler. "Sphere Drag and Settling Velocity Revisited". In: *Journal of Environmental Engineering* 129.3 (Mar. 1, 2003). Number: 3 Publisher: American Society of Civil Engineers, pp. 222–231. ISSN: 0733-9372. DOI: 10.1061/(ASCE)0733-9372(2003)129:3(222). URL: https://ascelibrary.org/doi/10.1061/%28ASCE%290733-9372%282003%29129%3A3%28222%29 (visited on 02/02/2024).

[17] Dominique Legendre et al. "A note on the bouncing of a drop or a solid sphere in viscous fluid". In: (Feb. 16, 2006).

[18] Mohammad Rostami et al. "Can the history force be neglected for the motion of particles at high subcritical Reynolds number range?" In: *International Journal of Engineering, Transactions B: Applications* 19 (Dec. 1, 2006).

[19] Weiming Wu and Sam S. Y. Wang. "Formulas for Sediment Porosity and Settling Velocity". In: *Journal of Hydraulic Engineering* 132.8 (Aug. 1, 2006). Number: 8 Publisher: American Society of Civil Engineers, pp. 858–862. ISSN: 0733-9429. DOI: 10.1061/(ASCE)0733-9429(2006)132:8(858). URL: https://ascelibrary.org/doi/10.1061/%28ASCE%290733-9429%282006%29132%3A8%28858%29 (visited on 02/08/2024).

[20] F.-L. Yang and M. L. Hunt. "Dynamics of particle-particle collisions in a viscous liquid". In: *Physics of Fluids* 18.12 (Dec. 8, 2006), p. 121506. ISSN: 1070-6631. DOI: 10.1063/1.2396925. URL: https://doi.org/10.1063/1.2396925 (visited on 03/04/2024).

[21] CIRIA; CUR Centre For Civil Engineering; CETMEF. *The Rock Manual. The use of rock in hydraulic engineering*. 2007.

[22] Heng Xiao and Jin Sun. "Algorithms in a Robust Hybrid CFD-DEM Solver for Particle-Laden Flows". In: *Communications in Computational Physics* 9.2 (Feb. 2011). Number: 2, pp. 297–323. ISSN: 1815-2406, 1991-7120. DOI: 10.4208/cicp.260509.230210a. URL: https://www.cambridge.org/core/product/identifier/S1815240600001134/type/journal_article (visited on 07/10/2023).

[23] N. Lukerchenko et al. "Drag Force, Drag Torque, and Magnus Force Coefficients of Rotating Spherical Particle Moving in Fluid". In: *Particulate Science and Technology* 30.1 (Jan. 1, 2012). Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/02726351.2010.544377, pp. 55–67. ISSN: 0272-6351. DOI: 10.1080/02726351.2010.544377. URL: https://doi.org/10.1080/02726351.2010.544377 (visited on 04/25/2024).

[24] T. N. Beemsterboer. "Modelling the immediate penetration of rock particles in soft clay during subsea rock installation, using a flexible fallpipe vessel." In: (2013). URL: https://repository.tudelft.nl/islandora/object/uuid%3A2dbc0c70-3280-4bcc-9245-1ea7a07fa773 (visited on 02/21/2023).

[25] M. D. J. Kevelam. "Hydrodynamics below a closed fallpipe system: The profile of a rock berm acquired with rock placement operations". In: (2016). URL: https://repository.tudelft.nl/islandora/object/uuid%3A5cb49711-a682-4e27-be50-eb2eec238e81 (visited on 01/17/2023).

[26] Huaqing Ma et al. "A review of recent development for the CFD-DEM investigations of non-spherical particles". In: *Powder Technology* 412 (Nov. 1, 2022), p. 117972. ISSN: 0032-5910. DOI: 10.1016/j.powtec.2022.117972. URL: https://www.sciencedirect.com/science/article/pii/S0032591022008531 (visited on 01/18/2023).

[27] B. J. Nieuwboer. "Modelling Spillage in Rotating Cutter Suction Heads: A combined Finite Volume and Discrete Element Model". In: (2022). DOI: 10.4233/uuid:ec0356e9-c264-4d1d-91b3-0a16ef13a348. URL: https://repository.tudelft.nl/islandora/object/uuid%3Aec0356e9-c264-4d1d-91b3-0a16ef13a348 (visited on 01/17/2023).

[28] *ANSYS FLUENT 12.0 Theory Guide - 15.2.2 Turbulent Dispersion of Particles*. Aug. 9, 2023. URL: https://www.afs.enea.it/project/neptunius/docs/fluent/html/th/node242.htm (visited on 08/09/2023).

[29] Boskalis. *Fallpipe vessel Rockpiper*. 2023.

[30] Jasper Fontijn. "Rock behavior in and below an inclined fall pipe". In: (2023). URL: https://repository.tudelft.nl/islandora/object/uuid%3A6efecd6a-d730-49c9-b3c6-46864aa0a95e (visited on 08/04/2023).

[31] Tim Hogenhuis. "Fallpipe dispersion model for subsea stone installation". In: (2023). URL: https://repository.tudelft.nl/islandora/object/uuid%3Af385b537-20b8-447b-b91b-190fdedb8baf (visited on 01/18/2024).

[32] *Magnus Effect*. Seatorque Control Systems. URL: http://www.seatorque.com/tech-info/magnus-effect/ (visited on 02/13/2024).

[33] *OpenFOAMAPI Guide*. URL: https://www.openfoam.com/documentation/guides/latest/api/StochasticDispersionRAS_8H_source.html (visited on 08/09/2023).

Colophon

This document was typeset using LaTeX, using the KOMA-Script class `scrbook`. The main font is Palatino.